Oracle Utilities
Customer to Meter

Administrative User Guide

Release 2.6.0.0

E86788-01

April 2017

Oracle Utilities Customer to Meter Administrative User Guide

Release 2.6.0.0

E86788-01

April 2017

Documentation build: 4.20.2017 21:40:00 [X1_1492738645000]

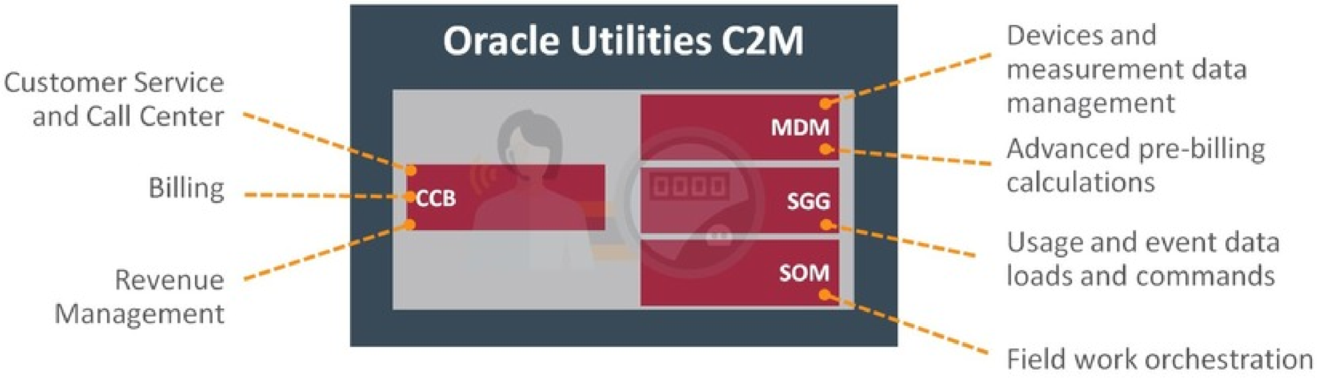
Contents

Customer to Meter Overview

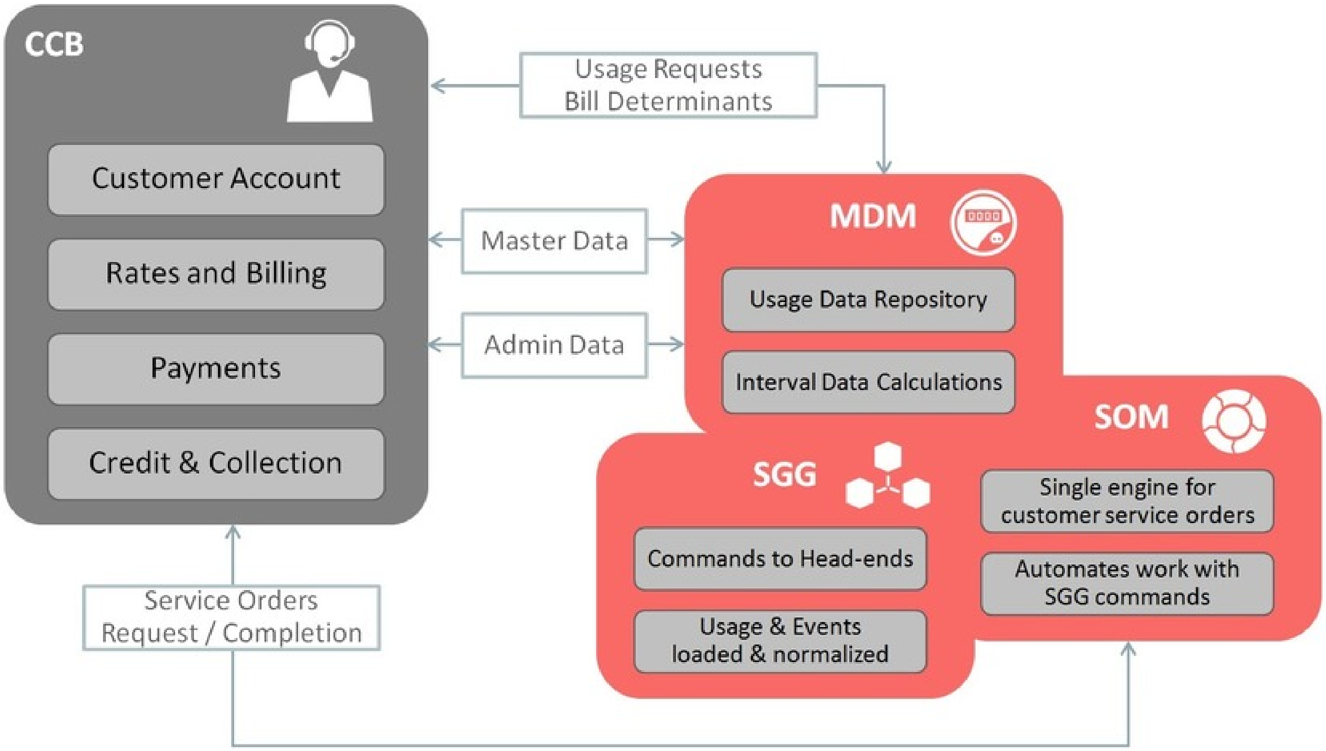
Customer to Meter Overview

Oracle Utilities Customer to Meter is a next generation customer service and billing application that incorporates a modern meter data management system. This system is designed to scale with mass smart meter volumes while handling the complexity associated with a utility's processes across multiple types of service, operating divisions, and jurisdictions. The new application components allow for traditional scalar meter functionality that is equivalent to prior versions of Oracle Utilities Customer Care and Billing; so, it can accommodate the many types of utility environments, and evolve with the business to include more advanced digital requirements.

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


Oracle Utilities C2M
Customer Service and Call Center
Billing
Revenue Management
CCB
MDM
SGG
SOM
Devices and measurement data management
Advanced pre-billing calculations
Usage and event data loads and commands
Field work orchestration

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



Meter Data Management Best Practices

Performance Recommendations

Initial Measurement Loading Recommendations

- In general, keep the number of devices to 1,000 per file for optimum usage and event processing through OSB. 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.
- 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.
- Populating the raw data section of the IMD record will reduce overall throughput

VEE Recommendations

- For the High-Low VEE Rule, the historic pre window should be set as low as possible based on business needs. This rule has an intensive hit on performance of the Initial Measurement loading process.

- 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 (0% - 5%)	Interval Interpolation Interval Replacement Rule Interval Size Validation Interval Spike Check Negative Consumption Check Scalar Replacement Rule UOM Check
Moderate Impact (5% - 20%)	Final Measurement Validation Interval Adjustment From Scalar Interval Averaging Multiplier Check Scalar Calculation From Interval Scalar Proration Sum Check
Significant Impact (20%)	High/Low Check

- 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.
- Using Scripting for simple VEE Rules is approximately 10% more expensive than a similar Java rule.
- 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.

Usage Transaction Recommendations

- 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 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.
- 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.

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.
- 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.

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.

Java Recommendations

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 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.

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 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.

Recommendations for Creating a 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 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 Configuration Migration Assistant to move system and administrative data. Instructions for using these tools are contained in the Bundling and Configuration Migration Assistant sections in the Oracle Utilities Application Framework *Administrative User Guide*.

Contact Oracle customer support if further assistance is required.

Additional Resources

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.
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

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
Information Center: Meter Data Management and Smart Grid Gateway Version 2	1907636.2
Related Measuring Component Consumption Sync White Paper	1672461.1
Overview and Guidelines for Managing Business Exceptions and Errors (this document was created by the CC&B team but pertains to MDM as well)	1628358.1

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 E-mail** 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.

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 right side Dashboard. 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 Configuration 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.

Customer To Meter Configuration

Oracle Utilities Customer to Meter employs functionality from both Oracle Utilities Customer Care and Billing and Oracle Utilities Meter Data Management. While much of the configuration of the application is very similar to that used with the individual applications, the merging of functions requires some configuration that is unique to Customer To Meter.

This chapter describes topics specific to configuring Customer To Meter to support a seamless integration between Customer Care and Billing and Meter Data Management functionality.

Customer To Meter Master Configuration

The Customer To Meter (C2M) master configuration provides a single point of access for configuration information related to consolidated data maintenance, data mapping, and other master configurations used with Customer To Meter.

To open the master configuration, select **Admin** > **General** > **C2M Configuration**. The master configuration opens in the C2M Configuration portal.

The **External System** field references the external system used for communication between Customer Care and Billing and Meter Data Management, including data synchronization requests, Service Order Management (SOM) queries, etc.

C2M Consolidated Maintenance

The **C2M Consolidated Maintenance** section documents objects that use some form of consolidated maintenance in Customer To Meter.

Merged Maintenance references the extendable lookup used to configure Merged Maintenance. Click **Joint Maintenance Lookup** to view values for this extendable lookup.

The **Consolidated Maintenance Objects** table lists objects that use Master Data Synchronization and Data Audit Maintenance for consolidated maintenance. Click on an object's name to view its maintenance object.

C2M Mapping

The **C2M Mapping** section lists objects used to map data between Customer Care and Billing and Meter Data Management.

SA Type Mapping references the extendable lookup used to map Customer Care and Billing service agreement (SA) types to Meter Data Management usage subscription types. Click **C2M SA Type Mapping Values** to view values for this extendable lookup.

The table lists fields that use Lookup Value Mapping to map values between Customer Care and Billing and Meter Data Management. Click the Description link to view and/or edit mapping values for a specific field. A plus sign (✚) indicates that mapping values have not been defined for that field. Click the icon to configure values for that field.

Related Master Configurations

The **Related Master Configurations** section lists other master configurations used with Customer To Meter, including:

- Master Data Synchronization Configuration
- Seeder Sync Request Master Configuration
- Service Order Management Master Configuration

- Service Order Management Integration

Click the **Master Configuration** link to view a specific master configuration.

Customer to Meter Consolidated Maintenance and Data Mapping

Oracle Utilities Customer to Meter uses data from both Oracle Utilities Customer Care and Billing and Oracle Utilities Meter Data Management. This chapter describes how data used by both in applications can be maintained through a consolidated maintenance process, and how data in Oracle Utilities Customer Care and Billing is mapped to similar data used by Oracle Utilities Meter Data Management.

There are several types of consolidated maintenance and data mapping used by Customer to Meter, including:

- Merged Maintenance: maintenance of data in both applications through use of a single portal.
- Master Data Synchronization: master data used by the two applications is kept synchronized when Customer Care Billing data is created or updated.
- Data Audit Maintenance: creating specific types of Meter Data Management data triggers creation of corresponding Customer Care and Billing data via Audit algorithms
- Lookup Value Mapping: lookup values used by Customer Care and Billing are mapped to corresponding Meter Data Management values via extendable lookups
- Transactional Data Mapping: creating specific types of transactional data in one part of Customer to Meter triggers the creation of corresponding transactions in the other part.

Merged Maintenance

Merged maintenance allows data in multiple product—specific maintenance objects to be updated via a single maintenance portal.

Administrative Data Merged Maintenance

Merged maintenance is used with several types of administrative data in Customer Care and Billing and Meter Data Management that are used in nearly identical ways in both applications. These include:

- Service Type
- SP Type (Service Point Type)
- Service Quantity Identifier
- Time of Use
- Unit of Measure

When a user creates or updates data of these types, the maintenance portal displays a superset of data elements from product-specific business objects based on a “superset” service script (defined in the Joint Maintenance Lookup extendable lookup (see below)). When the data is saved, the corresponding maintenance objects in both Customer Care and Billing and Meter Data Management are updated. For example, when a user navigates to the **Unit of Measure** portal and selects a record to update, the portal displays a combination of data from the X1-UnitOfMeasure and D1-UnitOfMeasure business objects. When the user saves the data, the changes are saved to both the UOM Customer Care and Billing maintenance object and the D1_UOM Meter Data Management maintenance object.

Service Point Merged Maintenance

In addition to the above types of administrative data, Customer to Meter service points are also maintained through merged maintenance. Service point merged maintenance does NOT make use of a “superset” service script, but does use product-specific business objects. Oracle Utilities Customer to Meter service points are maintained through these product-specific business objects and extendable lookups.

The Customer to Meter base package uses the following service point business objects:

- Meter Data Management: X1D-ServicePoint
- Customer Care and Billing: X1-ServicePoint

When a Customer to Meter service point is created or updated, the “Maintain CCB Service Point” Pre-Processing algorithm on the X1D-ServicePoint business object uses the C2M Mapping Field Values (X1–CCBMDM-MDMCCB-Lookup) extendable lookup to map lookup values between Customer Care and Billing and Meter Data Management, and creates or updates an instance of the X1–ServicePoint business object (which is used to update the SP maintenance object).

NOTE: Customer to Meter service point types should reference rhe X1D-ServicePoint business object.

Joint Maintenance Extendable Lookup

The specific business objects and service scripts used by merged maintenance are defined in the Joint Maintenance Lookup (X1-JointMaintenance) extendable lookup. Records in this extendable lookup define the superset script that contains the union of the elements of the objects that exist in Customer Care and Billing and Meter Data Management, as well as the corresponding product-specific business objects.

Merged maintenance lookup values are defined by the following:

- **CCB Maintenance Object**: the unique identifier for the joint maintenance lookup value
- **Superset Script**: the superset script that contains the union of the elements of the objects that exists in Customer Care and Billing and Meter Data Management. Used for administrative data maintenance only.
- **CCB Business Object**: the business object used to update the Customer Care and Billing maintenance object
- **MDM Business Object**: the business object used to update the Meter Data Management maintenance object

Customer To Meter Master Data Synchronization

This section describes the master data synchronization processes supported by Oracle Utilities Customer to Meter.

In Customer to Meter, Customer Care and Billing information is generally considered the data of record for account, customer, or service agreement-related data. Synchronizing master data between Customer Care and Billing tables and Meter Data Management tables ensures that all account, customer, and service point-related data is correct and up to date before usage transaction calculations are performed. Master data synchronization ensures that when specific types of Customer Care and Billing data are created or updated, corresponding Meter Data Management data is also created or updated.

The table below summarizes the types of data master data synchronization supported in Customer to Meter:

When Customer Care and Billing data of this type is created and/or updated:	Master Data Synchronization creates and/or updates this type of Meter Data Management data:
Bill Cycle / Bill Cycle Schedule	Bill Cycle / Bill Cycle Schedule
Contract Option	Dynamic Option
Contract Option Event	Dynamic Option Event
Person	Contact

When Customer Care and Billing data of this type is created and/or updated:	Master Data Synchronization creates and/or updates this type of Meter Data Management data:
Service Agreement	Usage Subscription

For example, when a Customer Care and Billing Service Agreement is created, the synchronization process creates a corresponding Meter Data Management usage subscription.

NOTE: Master data synchronization is a one-way process. When Customer Care and Billing data is created or updated, corresponding Meter Data Management data is created or updated, but changes made to Meter Data Management data are NOT synchronized to the Customer Care and Billing data. As a general rule, changes to the types of data outlined above should be made in the Customer Care and Billing entities instead of their Meter Data Management counterparts.

The Customer to Meter synchronization process is based on the Data Synchronization functionality of the Oracle Utilities Application Framework, and is supported through a set of maintenance objects, master configurations, business objects, and service scripts.

Maintenance Objects

Customer to Meter supports master data synchronization for the following maintenance objects.

Customer Care and Billing Maintenance Object	Meter Data Management Maintenance Object
BILL CYCLE (Bill Cycle)	D1-BILLCYCLE (Bill Cycle)
CONTRACT OPT (Contract Option)	D1-DOP (Dynamic Option)
CONTRACT OPT (Contract Option Event)	D1-DOPEVT(Dynamic Option Event)
PERSON (Person)	D1-CONTACT (Contact)
SA (Service Agreement)	D1–(Usage Subscription)

Master Configurations

The following master configurations are used to configure the sync process between Customer Care and Billing data and Meter Data Management data:

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. If initial synchronization process are also configured for data conversion in addition to the ongoing synchronization process, two views are specified. For entities that undergo the ongoing synchronization, 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 (contact, usage subscription, 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.

Outbound Data Synchronization Business Objects

When data is created for one of the above Customer Care and Billing maintenance objects, an outbound synchronization request is created to create the appropriate Meter Data Management data.

Synchronization of Customer Care and Billing data and Meter Data Management data uses the following outbound synchronization business objects:

Data Sync	Object Name	Business Object
Bill Cycle/Schedule > Bill Cycle/Schedule	MDM2 Bill Cycle Sync Request	C1-MDM2BillCycleSyncRequest
Contract Option > Dynamic Option	MDM2 Contract Option Sync Request	C1-MDM2ContractOptSyncRequest
Contract Option Event > Dynamic Option Event	MDM2 Contract Option Event Sync Request	C1-MDM2ContrOptEvtSyncRequest
Person > Contact	MDM2 Person Sync Request	C1-MDM2PersonSyncRequest
Service Agreement > Usage Subscription	MDM2 SA Sync Request	C1-MDM2SASyncRequest

Inbound Data Synchronization Service Scripts

When outbound synchronization requests enter the “Send Request” state, Enter algorithms invoke service scripts to create corresponding inbound data synchronization requests.

Synchronization of Customer Care and Billing data and Meter Data Management data uses the following inbound data synchronization service scripts:

Data Sync	Name	Service Script
Bill Cycle/Schedule > Bill Cycle/Schedule	Bill Cycle Inbound Sync Request	X1-InBCSynSS
Contract Option > Dynamic Option	Inbound Dynamic Option Sync	X1-InSyDyOp
Contract Option Event > Dynamic Option Event	Inbound Dynamic Option Event Sync	X1-InSyDyOpE
Person > Contact	Inbound Contact Sync	X1-InSyCont
Service Agreement > Usage Subscription	Inbound Usage Subscription Sync	X1-InSyUS

Inbound Data Synchronization Business Objects

When outbound synchronization requests enter the “Send Request” state, service scripts create corresponding inbound data synchronization requests. Master data synchronization uses two types of inbound synchronization requests:

- **Initial Synchronization Requests**: Initial synchronization requests are used during initial data conversion to facilitate creation of bill cycles/bill cycle schedules, dynamic options, dynamic options events, contacts, and usage subscriptions in Meter Data Management based on corresponding data in Customer Care and Billing. See Initial Data Synchronization for more information about initial synchronization requests
- **Ongoing Synchronization Requests**: Ongoing synchronization requests are used to continuously keep Meter Data Management data synchronized with corresponding Customer Care and Billing data. Ongoing synchronization requests can be used to maintain contacts, usage subscriptions, dynamic options, and bill cycles.

Ongoing synchronization of Customer Care and Billing data and Meter Data Management data uses the following data synchronization business objects:

Data Sync	Ongoing Synchronization Object Name	Business Object
Bill Cycle/Schedule > Bill Cycle/Schedule	Inbound Bill Cycle Sync Request	D2-InboundBillCycleSyncRequest

Contract Option > Dynamic Option	Dynamic Option Ongoing Sync Request	D2- OngoingSyncRequestDynOpt
Contract Option Event > Dynamic Option Event	Dynamic Option Event Ongoing Sync Request	D2- OngoingSyncRequestDynOptEvt
Person > Contact	Contact Ongoing Sync Request	D1-OngoingSyncRequestContact
Service Agreement > Usage Subscription	Usage Subscription Ongoing Sync Request	D2-OngoingSyncRequestUS

Example Synchronization Process Steps

This section provides an overview of the processing that takes place when an ongoing 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	Data in Customer Care and Billing is created or changed, triggering a synchronization request. For example, a service agreement is updated, and the corresponding usage subscription must be updated.	
2	An outbound synchronization request is created.	Outbound Synchronization Request business object: C1-MDM2SASyncRequest
3	When the synchronization requests enters the “Send Request” status, an Enter algorithm invokes the inbound synchronization service script to create an inbound sync request.	Inbound Synchronization Service Script: X1-InSyUS
4	The service script creates the appropriate inbound synchronization request. In this case, it’s an ongoing synchronization request.	Inbound Synchronization Request business object: D2-OngoingSyncRequestUS
5	For ongoing synchronization requests, the following steps are performed by Enter algorithms on the synchronization request business object lifecycle states: • Data Transformed/Basic Schema Validated • Determine Sync Request BO • Setup Transformed Data • Pre-Added • Sync Request Pre-Add Data • FKs Resolved • Resolve Keys - Ongoing Sync	Setup Transformed Data Algorithm: D1- SETTRANDT Sync Request Pre-Add Data Algorithm: D1-SR-PREADD Resolve Keys - Ongoing Sync Algorithm: D1-RESKEYFAL Sync Request Update Data Algorithm: D1- SR-UPDDAT

- Updating
- Sync Request Update Data

Initial Data Synchronization

This section describes the initial master data synchronization processes supported by Oracle Utilities Customer to Meter.

Initial data synchronization is used during initial data conversion to facilitate creation of data in Meter Data Management based on corresponding data in Customer Care and Billing. After creating bill cycles/schedules, contract options, contract option events, persons, and service agreements in Customer Care and Billing, initial data synchronization can be used to create corresponding bill cycles/schedules, dynamic options, dynamic options events, contacts, and usage subscriptions in Meter Data Management.

Initial data synchronization uses the same maintenance objects, master configurations, outbound synchronization request business objects, and service scripts as ongoing data synchronization (see Customer to Meter Master Data Synchronization for more information about these objects).

Inbound Initial Data Synchronization Business Objects

Initial synchronization of Customer Care and Billing data and Meter Data Management data uses the following data synchronization business objects:

Data Sync	Initial Synchronization Object Name	Business Object
Bill Cycle/Schedule > Bill Cycle/Schedule	Inbound Bill Cycle Sync Request	D2-InboundBillCycleSyncRequest
Contract Option > Dynamic Option	Dynamic Option Initial Sync Request	D2- InitialSyncRequestDynOpt
Contract Option Event > Dynamic Option Event	Dynamic Option Event Initial Sync Request	D2- InitialSyncRequestDynOptEvt
Person > Contact	Contact Initial Sync Request	D1-InitialSyncRequestContact
Service Agreement > Usage Subscription	Usage Subscription Initial Sync Request	D2-InitialSyncRequestUS

Initial Synchronization Request Batch Controls

Batch controls perform processing for initial synchronization requests such as allocating keys to data, resolving foreign keys, and loading data.

"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 (contact, usage subscription, etc.) to be processed. This must be set to the Sync Request maintenance object for the batch control (contact for contact synchronization requests, usage subscription for usage subscription 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.
- **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 (contact, usage subscription, etc,). Parameters used by "load data" batch controls include:

- **Maintenance Object**: (Required) the maintenance object (contact, usage subscription, etc.) to be processed. This must be set to the Sync Request maintenance object for the batch control (contact for contact synchronization requests, usage subscription for usage subscription 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-SIIER	Initial Sync Request - Error
D1-SILCN	Initial Sync Request - Load Data Contact
D1-SILUS	Initial Sync Request - Load Data US
D1-SIKCN	Initial Sync Request - Resolve Keys Contact
D1-SIKUS	Initial Sync Request - Resolve Keys US
D2-SIKUS	Initial Sync Request - Resolve Keys US
D2-SILUS	Initial Sync Request - Load Data US

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	Dynamic Option	Dynamic Option Event	Bill Cycle/Schedule
Transformation / Schema Validation Job	1	1	1	1	1
Key Allocation Job	2	2	2	2	2
Foreign Key Resolution / BO Validation Job (dependent on ALL Key Allocation Jobs finishing)	3	7	3	5	5
Load Job	4	8	4	6	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 contacts existing; Dynamic Option Event business object validation is dependent on Dynamic Options existing.

Example Initial Synchronization Process Steps

This section provides an overview of the processing that takes place when an initial 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	Data in Customer Care and Billing is created or changed, triggering a synchronization request. For example, several service agreements are created , and corresponding usage subscriptions must be created.	
2	An outbound synchronization request is created.	Outbound Synchronization Request business object: C1-MDM2SASyncRequest
3	When the synchronization requests enters the “Send Request” status, an Enter algorithm invokes the inbound synchronization service script to create an inbound sync request.	Inbound Synchronization Service Script: X1-InSyUS
4	The service script creates the appropriate inbound synchronization request. In this case, it’s an ongoing synchronization request.	Inbound Synchronization Request business object: D2-InitialSyncRequestUS
5	For initial synchronization requests, background processing creates master data for each synchronization request, including the following steps: • Data Transformation / Schema • Validation • Allocate Keys • Resolve Foreign Keys / Validate Business Object • Load Data	

Lookup Value Mapping

Oracle Utilities Customer to Meter service points and other records reference field and lookup values from both Oracle Utilities Customer Care and Billing and Oracle Utilities Meter Data Management. This section describes how field and lookup values used by Customer Care and Billing are mapped to similar field and lookup values used by Oracle Utilities Meter Data Management.

Mapping of field and lookup values is supported through the use of two extendable lookups:

- **C2M Mapping Field List** (X1-CCBMDM-MDMCCB-MappingField): used to capture the fields that require translation mapping between Customer Care and Billing and Meter Data Maintenance maintenance objects in Customer to Meter. Values in this extendable lookup are identified by a field or lookup used by Customer Care and Billing, such as Disconnect Location (DISCON_LOC_CD) or Life Support / Sensitive Load (LS_SL_FLG).

- **C2M Mapping Field Values** (X1–CCBMDM-MDMCCB-Lookup): used to capture the mapping values for the fields and lookups defined in the C2M Mapping Field List extendable lookup. Values in this extendable lookup are identified by a Customer Care and Billing field or lookup defined in the C2M Mapping Field List extendable lookup. Each value also defines the specific value mappings for the field or lookup.

Examples

Value Mappings for the Disconnect Location (DISCON_LOC_CD) field are as follows:

CCB Value	MDM Value
Disconnected at meter	Device
Disconnected at pole	Source

Value Mappings for the Life Support / Sensitive Load (LS_SL_FLG) lookup are as follows:

CCB Value	MDM Value
None	Not on life support
LS/SL	On life support

Data Audit Maintenance

Different functions in Customer Care and Billing and Meter Data Management use similar types of data when calculating usage and billing charges, and it’s critical that the data be consistent between the applications. For example, both Usage Rules in Meter Data Management and Calculation Rules in Customer Care and Billing make use of TOU Groups, TOU Map Templates, TOU Map Types, etc., and when calculating usage and billing charges for an account, the data used by both systems should be effectively identical. The process of maintaining similar sets of data usable by each system is called Data Audit Maintenance.

When specific types of Meter Data Management data is created or updated, Audit algorithms create or update corresponding Customer Care and Billing data. For instance, when a TOU Map Type usable by Meter Data Management Usage Rules is created, data auditing creates a similar TOU Map Type usable by Customer Care and Billing Calculation Rules.

Data audit maintenance is used with several types of Meter Data Management data in Customer to Meter, including:

- TOU Group
- TOU Map Template
- TOU Map Type
- TOU Map
- Device Type (Item Type)

The table below outlines the business objects and algorithms used by data audit maintenance:

Data Type	Business Object	Audit Algorithm
TOU Group	D2-TOUGroup	Maintain CCB TOU Group (X1-MCCBTOUGP)
TOU Map Template	D2-TOUMapTemplate	Maintain CCB TOU Map Template (X1-MCCBTOUTM)
TOU Map Type	D2-TOUMapType	Maintain CCB TOU Map Type (X1-MCCBTOUMT)
TOU Map	D2-TOUMap	Maintain CCB TOU Map (X1-MCCBTOUMP)
Device Type (Item Type)	D1-ItemType	Maintain CCB Item Type (X1-MCCBITEMT)

Refer to the Business Object and Algorithm portals to view additional details regarding these audit algorithms.

Other Data Auditing

In addition to the objects listed above, Customer To Meter uses data auditing with other objects in Customer Care and Billing to ensure that changes to data reflected in related data in Meter Data Management. These objects use Audit algorithms on their maintenance objects to capture any changes and update the corresponding data. The objects that make use of this approach include the following:

- **Account**: When service agreement-related account data is changed in Customer Care and Billing, the audit algorithm creates a Service Agreement outbound data synchronization request to captures the change and update the corresponding usage subscription in Meter Data Management.
- **Premise**: When a Customer Care and Billing premise is changed, the audit algorithm updates the address information in the service point in Meter Data Management.

The table below outlines the data types, maintenance objects, and algorithms used by these.

Data Type	Maintenance Object	Audit Algorithm
Account	ACCOUNT	C2M Account Change Data Capture (SA-Based) (X1-ACCTCDCSA)
Premise	PREMISE	Update MDM SP from CCB Premise (X1-UPSPFRPR)

Refer to the Maintenance Object and Algorithm portals to view additional details regarding these audit algorithms.

Transactional Data Mapping

Certain types of transactional data created in Customer Care and Billing result in related transactional data in Meter Data Management (and vice-versa). This section describes these transactional mappings.

Usage Requests / Usage Transactions

Customer Care and Billing creates usage requests during the billing process to request usage calculations in Meter Data Management. These requests in turn create Meter Data Management usage transactions, which perform the usage calculations.

Customer Care and Billing usage requests use the following business objects:

- MDM Cyclical Usage Request (C1–UsageRequestCyclicalBilling): Used with usage requests created from the cyclical or batch billing process.
- MDM Non-Cyclical Usage Request (C1–UsageRequestNonCyclical): Used with usage requests created from non-cyclical or online bill generation.
- MDM Usage Request (C1–UsageRequest): Parent business object that defines the lifecycle for Cyclical and Non-Cyclical usage requests.

When a usage request enters the “Send Request” state, the “Send Usage Request Message to MDM” (X1–SENDUSGR) Enter algorithm invokes the “Usage Transaction Request Inbound” (X1-UsgTrRqIn) service script which uses the Usage Transaction Seeder (D1–UsgTranSeeder) business object to create a usage transaction.

When a Meter Data Management usage transaction enters a “Subsequent Correction” state, the “X1-SEND-SUBC” Enter algorithm invokes the “Call Subsequent Correction Notification” (X1-SendSubC) service script to process a corrected read notification for Customer Care and Billing by finding the account associated with the corrected read and creating an off cycle bill generator to handle associated correction processing.

Refer to The Big Picture of Usage Requests for more information about usage requests and usage transactions.

Field Activity Requests / Service Order Activities

Customer Care and Billing creates Field Activity Requests to initiate service orders to perform tasks at a service point. Most field activity requests are created:

- When a customer service representative starts or stops service at a premise. When this happens, the system automatically creates field activity requests for each service point.
- When service is cut due to lack of payment or reconnected for sufficient payment or arrangement.

When Customer Care and Billing creates a field activity request, a corresponding Service Order Activity is created to orchestrate the various activities required to fulfill the service order by continuously monitoring the status of the service point, device installation, and related measurements. This orchestration can include field activities to instruct a field crew to install, remove, or exchange a meter, as well as eligible smart meter command activities to connect or disconnect the meter, etc.

Service Order Field Activities are specific activities that typically exist only in Service Order Management and not Customer Care and Billing. However, field activity remarks notification, appointment notification (if required by the service order field activity), a missed appointment (which can create an adjustment) and create customer contact actions can be invoked in Customer Care and Billing when triggered from service order field activities.

Refer to Understanding Service Order Management for more information about service order activity processing.

Customer To Meter Table Setup Sequence

This section provides guidelines for setting up control tables and administrative data used with Oracle Utilities Customer To Meter.

Customer Care and Billing Control Table Setup Sequence

The topics in this section describe the order in which the control tables should be set up.

Core Control Table Setup Sequence

To implement the system, you must set up your organization's business rules in "control tables". Setting up these tables is time-consuming because we allow you to tailor many aspects of the system to meet your organization's requirements. We strongly recommend that you take the time to document how you plan to set up all of these tables before you use the following roadmap to enter the core control data. Time spent understanding the interrelationships between this data will reap the rewards of a clean system that meets your current and long term needs.

While we describe the transactions and options in more detail in other sections of this manual, use the following chart (and the remaining sections of this chapter) as your roadmap. Here we list the order in which you perform tasks and the pages you'll use to set up your system. The order is important because some information must exist before other information can be defined (i.e., many dependencies exist).

➤ **NOTE: Auto setup.** The Auto Setup column in the following table contains suggestions to save you time. It also indicates if a control table contains information when the system is installed.

➤ **NOTE: You don't have to set up every control table.** You need only set up those control tables that govern functions that are applicable to your organization.

Function	Menu	Auto Setup
Global Context		
Algorithm	Admin, Algorithm. You will need to set up an algorithm that populates global context values. The global context is used by various zones in the system to display relevant data. This algorithm is plugged-in on the installation record.	
Accounting Environment		
Country and State	Admin, Country	
Currency Codes	Admin, Currency Code	USD is automatically populated
Accounting Calendar	Admin, Accounting Calendar	
GL Division	Admin, General Ledger Division	
Security Environment		
Application Service	Admin, Application Service	All base package transactions are automatically populated
Security Type	Admin, Security Type	
User Group	Admin, User Group Note, you won't be able to set up users at this point	One user group, ALL_SERVICES, is automatically setup. It references all other application services and a single user called SYSUSER.
Language	Admin, Language	ENG is automatically populated
Display Profile	Admin, Display Profile	Two display profiles are automatically setup: NORTHAM displays currencies and dates in a classic American format; EURO displays information in a classic European format
Data Access Role	Admin, Data Access Role	
Access Group	Admin, Access Group	
User	Admin, User	SYSUSER is automatically set up. Note that you may import your users (and user groups) from an external source.
Return to User Group	You must return to your user groups and define all of their users	
Customer Class Environment		
Customer Class	Admin, Customer Class. At this point, you'll only be able to set up your customer class codes. You will return to these customer classes throughout the setup process to populate additional information.	
Financial Transaction Environment		

Work Calendar	Admin, Work Calendar
CIS Division	Admin, CIS Division
Revenue Class	Admin Revenue Class
Algorithm	Admin, Algorithm. You will need to set up the algorithm that constructs a distribution code's corresponding GL account when it is interfaced to the general ledger
Distribution Code	Admin, Distribution Code
Bank and Bank Accounts	Admin, Bank
Billable Charge Template	Admin, Billable Charge Template. Note, if you want the system to default service quantities onto billable charges created using this template, you must setup the appropriate unit of measure code, time-of-use code and/or service quantity identifier.
Billable Charge Upload Line Type	Admin, Billable Charge Line Type
Algorithm	Admin, Algorithm. You will need to set up several algorithms. These algorithms: 1) retrieve a bill segment's consumption, 2) calculate a bill segment's bill lines, 3) construct a bill segment's financial transaction, 4) cancel previously estimated bill segments
Bill Segment Type	Admin, Bill Segment Type
Algorithm	Admin, Algorithm. You may want to set up an algorithm that formats the Bill Segment information that is displayed throughout the system for a specific Bill Segment Type. This algorithm is plugged-in on the Bill Segment Type.
Algorithm	Admin, Algorithm. You will need to set up the algorithm that constructs a payment segment's financial transaction
Payment Segment Type	Admin, Payment Segment Type
Algorithm	Admin, Algorithm. You will need to set up the algorithm that constructs an adjustment's financial transaction
Algorithm	Admin, Algorithm. Several plug-in spots are available to perform additional logic when processing adjustments. For example, if you have the system calculate adjustments, you must set up an adjustment generation algorithm. Refer to Adjustment Type for other available plug-in spots that may be used by your implementation.
Algorithm	Admin, Algorithm. You may want to set up an algorithm that formats the Adjustment information that is displayed throughout the

	system for a specific Adjustment Type. This algorithm is plugged-in on the Adjustment Type.
Algorithm	Admin, Algorithm. You may want to set up an algorithm that formats the Adjustment information that is displayed throughout the system. This algorithm is plugged-in on the installation record.
Adjustment Type	Admin, Adjustment Type
Adjustment Type Profile	Admin, Adjustment Type Profile
Approval Profile	Admin, Approval Profile. Note, an approval profile references a To Do type and one or more To Do Roles; these must be set up before you can set up the approval profile. After the approval profile(s) are set up, they must be referenced on the adjustment types that they govern.
Cancel Reason - Bill	Admin, Bill Cancel Reason
Cancel Reason - Payment	Admin, Payment Cancel Reason
Cancel Reason - Adjustment	Admin, Adjustment Cancel Reason
Tender Type	Admin, Tender Type
Tender Source	Admin, Tender Source
A/P Request Type	Admin, A/P Request Type
Issuing Center	Admin, Issuing Center. You will need to set up issuing centers if your organization assigns document numbers to bills.
Installation	Admin, Installation Options - Framework and Admin, Installation Options. Many fields on the installation record impact the financial transaction environment. Refer to the description of the Billing and Financial Transaction tabs and the Messages tab in the Framework page for more information.
Algorithm	Admin, Algorithm. You will need to set up an algorithm that distributes payments.
Algorithm	Admin, Algorithm. You will need to set up an algorithm that handles overpayment situations.
Algorithm	Admin, Algorithm. You may need to set up an algorithm if specific customers can have individual bill due dates.
Algorithm	Admin, Algorithm. You may need to set up an algorithm if you want the system to delete bills that contain only information about historical payments.
Algorithm	Admin, Algorithm. You may need to set up an algorithm if you want the system to levy

	a non-sufficient funds charge if a payment is canceled due to non-sufficient funds.
Algorithm	Admin, Algorithm. You will need to set up an algorithm that formats the bill information that is displayed throughout the system. This algorithm is plugged-in on the installation record.
Algorithm	Admin, Algorithm. You will need to set up an algorithm that formats the payment information that is displayed throughout the system. This algorithm is plugged-in on the installation record.
Algorithm	Admin, Algorithm. You will need to set up an algorithm that defaults the amount when a payment is manually added. This algorithm also calculates the amount of an automatic payment for a bill for an account with an active auto pay option. This algorithm is plugged-in on the installation record.
Algorithm	Admin, Algorithm. Refer to Customer Class for other available plug-in spots that may be used by your implementation to perform additional logic when processing payments and bills.
Return to Customer Class	Admin, Customer Class. You will need to plug-in the algorithms defined above on your customer classes.
Budget Environment	
Algorithm	Admin, Algorithm. You will need to set up several algorithms at this time: How To Calculated The Recommended Budget Amount, How To Periodically True Up A Customer's Budget Amount, The Circumstances When The System Should Highlight A Customer As Having An Anomalous Budget.
Budget Plan	Admin, Budget Plan
Algorithm	Admin, Algorithm. Budget eligibility is set at the SA type level. You will need to set up an override budget eligibility algorithm if some service agreements for an SA type are not eligible for budget based on certain conditions.
Customer Environment	
Account Management Group	Admin, Account Management Group. Note, you will probably have to set up To Do Type and To Do Roles before you can setup account management groups. Refer to Assigning A To Do Role for more information on how account management groups may be used to define an entry's role.

Account Relationship	Admin, Account Relationship Type
Alert Type	Admin, Alert Type
Bill Message	Admin, Bill Message
Algorithm	Admin, Algorithm. If you have software that's capable of reconstructing an image of a bill in a PDF (for the purpose of online display), you will need to create an algorithm that formats the extract records that are sent to your bill image software.
Bill Route Type	Admin, Bill Route Type
Contract Quantity Type	Admin, Contract Quantity Type
Algorithm	Admin, Algorithm. If you have software that's capable of reconstructing an image of a letter in a PDF (for the purpose of online display), you will need to create an algorithm that formats the extract records that are sent to your letter image software.
Letter Template	Admin, Letter Template
Customer Contact Class	Admin, Customer Contact Class
Customer Contact Type	Admin, Customer Contact Type
Conservation Programs	Admin, Conservation Program. You will need to set up conservation programs if your organization provides rebates to customers based on eligibility and verification of newly purchased appliances and hardware that are rated to conserve the demand for energy.
Algorithm	Admin, Algorithm. You may need to set up the algorithms that determine if person ID's are in a predefined format.
Identifier Type	Admin, Identifier Type
SICs	Admin, SIC Code
Tax Exempt Type	Admin, Tax Exempt Type
Algorithm	Admin, Algorithm. You may need to set up the algorithms that determine if phone numbers are in a predefined format.
Phone Type	Admin, Phone Type.
Person Relationship Type	Admin, Person Relationship Type.
Algorithm	Admin, Algorithm. You will need to set up an algorithm that formats the person information that is displayed throughout the system. This algorithm is plugged-in on the installation record.
Algorithm	Admin, Algorithm. You will need to set up an algorithm to validate a person's name. This algorithm is plugged-in on the installation record.

Algorithm	Admin, Algorithm. You can override the system's standard account information string by setting up an algorithm that produces this string of information. This algorithm is plugged-in on the installation record.
Algorithm	Admin, Algorithm. If you have software that's capable of reconstructing an image of a bill in a PDF for the purpose of online display, you will need to create an algorithm that renders this PDF. This algorithm is plugged-in on the installation record.
Installation	Admin, Installation Options. Many fields on the installation record impact the Customer Environment. Refer to the description of the Main, Person, and Account tabs for more information.
Statements	
Algorithm	Admin, Algorithm. If you have software that's capable of reconstructing an image of a statement in a PDF (for the purpose of online display), you will need to create an algorithm that formats the extract records that are sent to your statement image software.
Statement Route Type	Admin, Statement Route Type
Statement Cycle	Admin, Statement Cycle
Algorithm	Admin, Algorithm. If you have software that's capable of reconstructing an image of a statement in a PDF for the purpose of online display, you will need to create an algorithm that renders this PDF. This algorithm is plugged-in on the installation record.
Automatic Payment (EFT) Environment	
Algorithm	Admin, Algorithm. You will need to set up an algorithm to create automatic payments. This algorithm is plugged-in on the installation record.
Tender Source	Admin, Tender Source Note: earlier, you created tender sources for the remittance processor and your cash drawers. At this point, you'll need to add at least one tender source for automatic payments. Why? Because automatic payments get linked to a tender control (which, in turn, gets linked to a tender source) when they are interfaced out of the system.
Algorithm	Admin, Algorithm. You will need to set up the appropriate automatic payment date calculation algorithm to populate the extract,

	GL interface and payment dates on automatic payments.
Auto Pay Route Type	Admin, Auto Pay Route Type
Tender Type	Admin, Tender Type Note: earlier, you created tender types for things like cash, checks, etc. At this point, you'll need to add a tender type for each type of automatic payments (e.g., direct debt, credit card, etc.).
Work Calendar	Admin, Work Calendar. You need only set up additional work calendars if the auto pay sources (i.e., the financial institutions) have different working days than does your organization
Algorithm	Admin, Algorithm. If you need to validate the customer's bank account or credit card number, you will need to set up the appropriate validation algorithms.
Auto Pay Source Type	Admin, Auto Pay Source Type
Algorithm	Admin, Algorithm. You may need to set up an algorithm if your customers can define a maximum withdrawal limit on their autopay options.
Return to Customer Class	Admin, Customer Class. You should plug-in the Autopay Over Limit Algorithm in each appropriate customer class.
Deposit Environment	
Algorithm	Admin, Algorithm. You will need to set up several algorithms at this time: The Definition Of A Good Customer, When To Refund A Deposit To A Customer, When To Recommend An Additional Deposit, How / When To Calculate Interest, How To Generate The Recommended Deposit Amount.
Deposit Class	Admin, Deposit Class
Non Cash Deposit Type	Admin, Non Cash Deposit Type
Field Work Environment	
Field Activity Type and Steps	Admin, Field Activity Type
Field Activity and Field Order Cancellation Reason	Admin, Fieldwork Cancel Reason
Field Activity Type Profiles	Admin, Field Activity Type Profile
Algorithm	Field activities for Start/Stop will only be created if the SA Type linked to the service point has a SASP field work creation algorithm. Refer to the SA Type section below.

Algorithm	Admin, Algorithm. If you have software that's capable of reconstructing an image of a field order in a PDF for the purpose of online display, you will need to create an algorithm that renders this PDF. This algorithm is plugged-in on the installation record.
Credit & Collections Environment (if you collect on overdue bills (as opposed to overdue debt), you will NOT set up these tables; refer to Overdue Processing - Set Up Tasks for the list of control tables required to collect on overdue bills)	
Algorithm	Admin, Algorithm. You may need to set up algorithms if you have non-standard collection events.
Collection Event Type	Admin, Collection Event Type
Algorithm	Admin, Algorithm. You may need to set up a collection process cancellation algorithm if your organization allows individual service agreements to be removed from a collection process if they are paid (rather than performing cancellation based on all SAs in a debt class).
Collection Process Template	Admin, Collection Process Template
Collection Class	Admin, Collection Class
Algorithm	Admin, Algorithm. You will need to set up several algorithms at this time: Collection process cancellation criteria, Severance process cancellation criteria, and Override arrears due to pay plans.
Debt Class	Admin, Debt Class
Write Off Debt Class	Admin, Write Off Debt Class
Algorithm	Admin, Algorithm. You will need to set up Collection Condition algorithms.
Collection Class Control	Admin, Collection Class Control
Algorithm	Admin, Algorithm. You may need to set up algorithms if you have non-standard severance events.
Severance Event Type	Admin, Severance Event Type
Algorithm	Admin, Algorithm. You may need to set up a severance process cancellation algorithm if your organization allows a severance process to be canceled when the related service agreement is paid (rather than performing cancellation based on all SAs in a debt class).
Severance Process Template	Admin, Severance Process Template
Algorithm	Admin, Algorithm. You will need to set up several algorithms at this time: How to refer debt to a collection agency, How to transfer debt to another active service agreement, How to write down small amounts of debt, and How to refund credit balances to a customer.

Algorithm	Admin, Algorithm. You may need to set up algorithms if you have non-standard write-off events.	
Write Off Event Type	Admin, Write Off Event Type (Note, you'll have to wait until you have defined your SA Types before you can set up the Write Off Events because SA Type is a necessary parameter to write off debt).	
Write Off Process Template	Admin, Write Off Process Template	
Write Off Control	Admin, Write Off Control	
Collection Agency	Admin, Collection Agency. Note, each collection agency references a person therefore you must set up a person for each agency before you can enter collection agency information.	
Algorithm	Admin, Algorithm. You may need to set up algorithms if you have special logic that should be executed when a pay plan is canceled.	
Pay Plan Type	Admin, Pay Plan Type	
Payment Method	Admin, Payment Method	
Third Party Payor	Admin, Third Party Payor. Note, you must create an account before you can create a third party payor.	
Installation	Admin, Installation. Several fields on the installation record impact the Credit & Collections Environment.	
Algorithm	Admin, Algorithm. You will need to setup an algorithm that's called when a user write-off debt real time.	
Return to Customer Class	Admin, Customer Class. You should plug-in the Autopay Over Limit Algorithm in each appropriate customer class.	
Services and Characteristics		
Service Type	Admin, Service Type	
Algorithm	Admin, Algorithm. If you have ad hoc characteristic types, you may need to set up the algorithms that control how they are validated	
Foreign Key Reference	Admin, FK Reference. If you have foreign key characteristic types, you may need to set up foreign key references to control how the user selects the characteristic values (and how the foreign key values are validated).	All base package FK references are automatically populated
Characteristic Type and Values	Admin, Characteristic Type	
Meter and Item Environment	Refer to the Meter Data Management Table Setup Sequence for control tables used with Meter Data Management, and configure the control tables as specified in this section.	

Premise and Service Point Environment		
Premise Type	Admin, Premise Type	
Algorithm	Admin, Algorithm. You will need to set up an algorithm to format the standard premise info that appears throughout the system. This algorithm is plugged-in on the installation record.	
Algorithm	Admin, Algorithm. You will need to set up an algorithm that formats the service point information that is displayed throughout the system. This algorithm is plugged-in on the installation record.	
Algorithm	Admin, Algorithm. You may need to set up the algorithms that determine if geographic ID's are in a predefined format.	
Geographic Type	Admin, Geographic Type	
Service Point Type	Admin, SP Type. [Note - you won't be able to define the SP Type's SA Types until after you define the SA Types or the FA Type Profiles until after you define the Field Activity Type Profiles.]	
Bill and Service Cycle Environment		
Bill Cycle, Bill Cycle Schedule	Admin, Bill Cycle	
Bill Period, Bill Period Schedule	Admin, Bill Period	
Rate Environment		
Frequency	Admin, Frequency	
Service Quantity Identifier	Admin, Service Quantity Identifier	
Algorithm Type	Admin, Algorithm Type. If you create new pre-processing calculation rules, you must set up an algorithm type for each such rule (the algorithm type defines the types of parameters that are passed to the calculation rule).	All base package algorithm types are automatically populated
Bill Factor	Main, Rates, Bill Factor	
Algorithm Type	Admin, Algorithm Type. If you create new Register Rules you must set up an algorithm type for each such rule (the algorithm type defines the types of parameters that are passed to the register rule).	All base package algorithm types are automatically populated
Rate	Main, Rates, Rate Schedule	
Calculation Group	Main, Rates, Calculation Group	
Algorithm	Admin, Algorithm. If you use algorithms to dynamically change step boundaries, calculate prices, or implement calculation rule eligibility rules, you must set up these algorithms.	
Calculation Rule	Main, Rates, Calculation Rule	

Bill Factor Value	Main, Rates, Bill Factor Values
Bill Factor Interval Values	Main, Rates, BF Interval Values
Item Type SQ Estimate	Admin, Item Type SQ Estimate
Degree Day	Admin, Degree Days
Late Payment Environment	
Algorithm	Admin, Algorithm. You will need to set up the algorithm that determine if customers in a customer class are eligible for late payment charges
Algorithm	Admin, Algorithm. You will need to set up the algorithm that levies late payment charges for customers in a customer class
Return to Customer Class	Admin, Customer Class. You will need to plug-in the late payment charge algorithms set up above.
SA Configuration	
Algorithm	Admin, Algorithm. You will need to set up the algorithms that determine: • How to calculate the late payment charge amount for service agreements of a given type • Special criteria to be tested before a service agreement is severed. • How to create field activities for service agreements of a given type. • Special processing that should take place prior to the completion of a bill that references service agreements of a given type. • Special processing that should take place during completion of a bill that references service agreements of a given type. • Special processing that should take place when service agreements of a given type are created. • Special processing that should take place when a financial transaction is frozen for service agreements of a given type.
Algorithm	Admin, Algorithm. You may want to set up an algorithm that formats the SA information that is displayed throughout the system. This algorithm is plugged-in on the installation record.
Algorithm	Admin, Algorithm. You may want to set up an algorithm that formats the SA information that is displayed throughout the system for a specific SA Type. This algorithm is plugged-in on the SA Type.

Algorithm	Admin, Algorithm. If you want a Control Central alert to highlight when the current account has any stopped service agreement(s), you will need to set up the algorithm that does this. This algorithm is plugged-in on the installation record.
Service Agreement Type	Admin, SA Type
Terms and Conditions	Admin, Terms and Conditions
SA Type Start Options	Admin, SA Type Start Option
Update SP Types with initial SA types and with FA Type Profiles	Admin, SP Type
SA Relationships	
SA Relationship Type	Admin, SA Relationship Type
Service Provider	Admin, Service Provider. Note, you must create a person before you can create a service provider. If you have financial relationships (you bill for them or they bill for you), you must also create an account and a financial settlement service agreement before you can create the service provider.
SA Type / SA Relationship	Admin, SA Type SA Relationship Type
Service Provider.	Admin, Service Provider. Note, you must create a person before you can create a service provider.
Sales and Marketing	
Order Hold Reason	Admin, Order Hold Reason
Order Cancel Reason	Admin, Order Cancel Reason
And more...	Refer to Campaign and Package Setup Sequence for additional setup requirements
Service Credit Membership	
Algorithm	Admin, Algorithm. You may need to set up algorithms for the service credit membership type and service credit event type to control behavior for the service credit membership and its events.
Credit Unit	Admin, Credit Unit. If your service credits record non-monetary units.
Service Credit Membership Type	Admin, Service Credit Membership Type
Service Credit Event Type	Admin, Service Credit Event Type
Membership Inactive Reasons	Admin, SC Membership Inactive Reason
Wrap Up	
Algorithm	Admin, Algorithm. You will need to set up the algorithms that determine: #x2022; Special alerts on Control Central (assuming you have special alerts)

Installation Options	Admin, Installation Options - Framework and Admin, Installation Options. At this point, it's a good idea to double-check everything on the installation record.
Postal Default	Admin, Postal Code Default

If you have cash drawers you will also need to set up the following information:

- Create a person / account to which you will link your over / under service agreement. Refer How To Get An Unbalanced Tender Control In Balance (Fixing Over/Under) for more information.
- Create a service agreement to which your over/under payments will be linked. This service agreement will reference your over / under SA type. Refer to Over / Under Cash Drawer Segmentation for more information.

If you upload payments from an external source (e.g., a remittance processor or lock box), you must set up the following information:

- Create a person and account to which the system will link payments with invalid account. Refer to Phase 3 - Create Payment Events, Tenders, Payments and Payment Segments for information about the process that books invalid payments to this account. Refer to How To Transfer A Payment From One Account To Another for how a user transfers the payment from the invalid account to the correct account (once known).
- Create a service agreement for this account. This service agreement will reference your payment suspense SA type. The system needs this service agreement so that it can distribute the invalid account's payment (and this is necessary so that cash will reflect the payment). Refer to Payment Upload Error Segmentation for more information.
- Update the tender source associated with the respective source of payments to indicate the service agreement created in the previous step should be used for payments with invalid accounts. Refer to Setting Up Tender Sources for more information.
- Because the payment upload process simply books payments that reference invalid accounts to the account associated with the suspense service agreement on the payment's tender source, this account should belong to a customer class with the appropriate payment distribution algorithms. This may entail creating a new customer class that will only be used on suspense accounts. This customer class would need the following algorithms:
- We'd recommend using a simple payment distribution algorithm like PYDIST-PPRTY (distribute payment based on SA type's payment priority).
- We'd recommend using an overpayment distribution algorithm like OVRPY-PPRTY (distribute overpayment to highest priority SA type).

The remaining sections describe additional control tables that must be set up for specific functional areas.

Cross-Reference To The Remaining Chapters

The table in the previous section describes the order in which you should enter your control tables. These tables are described at length in the following chapters.

- Refer to Defining General Options Addendum and Defining General Framework Options for a discussion of the control tables associated with general functionality (e.g., country codes, state codes, etc.).
- Refer to Defining Financial Transaction Options for a discussion of the tables affecting your financial transactions (e.g., bill segment types, payment segment types, etc.)
- Refer to Defining Customer Options for a discussion of the control tables affecting persons, accounts and service agreements.
- Refer to Defining Service Order Options for a discussion of the control tables affecting fieldwork.
- Refer to Defining Credit and Collections Options for a discussion of the control tables affecting your collection activities.

- Refer to Defining Device Options for a discussion of the control tables affecting your meters and items.
- Refer to Defining Device Installation Options for a discussion of the control tables affecting your premises and service points.
- Refer to Defining Cycles and Schedules for a discussion of the control tables affecting your cyclical processes.
- Refer to Rates for a discussion of the control tables affecting your rates.
- Refer to Defining SA Type Options for a discussion of the control tables affecting your service agreement types.
- Refer to Defining Background Process for a discussion of the control tables affecting your background processes.
- Refer to Defining Algorithms for a discussion of the control tables affecting the algorithms referenced on many control tables.
- Refer to Defining SA Relationships for a discussion of the control tables affecting the relationships between service providers.
- Refer to Statements for a discussion of the tables affecting the statement setup options for your customers.
- Refer to Defining Service Credit Options for a discussion of the tables affecting the service credit membership setup options for your customers.

Open-Item Accounting Table Setup Sequence

Open-item accounting tables need only be set up if your organization practices Open Item Accounting. Refer to Setting Up The System To Enable Open Item Accounting for a description of the tables that must be set up to enable this functionality.

Fund Accounting Table Setup Sequence

Fund accounting tables need only be set up if your organization practices Fund Accounting. Refer to Setting Up The System To Enable Fund Accounting for a description of the tables that must be set up to enable this functionality.

Payment Event Distribution Table Setup Sequence

Payment event distribution tables need only be set up if your organization opted to use the distribution rules method to create payment events. Refer to Setting Up The System To Use Distribution Rules for a description of the tables that must be set up to enable this functionality.

Loans Table Setup Sequence

Loans need only be set up if your organization offers loans to your customers. Refer to Setting Up The System To Enable Loans for a description of the tables that must be set up to enable this functionality.

Quotes Table Setup Sequence

Quotes need only be set up if your organization sends quotes to customers or prospects. Refer to Defining Quotation Options for a description of the tables that must be set up to enable this functionality.

Non-Billed Budget Table Setup Sequence

Non-Billed Budgets need only be set up if your organization allows your customers to pay set amounts at specified intervals (e.g. every two weeks). Refer to Setting Up The System To Enable Non-Billed Budgets for a description of the tables that must be set up to enable this functionality.

Scripting Table Setup Sequence

Scripts need only be set up if your organization opts to create scripts to step your users through business processes. Refer to Defining Script Options for information about scripting and the tables that must be set up to enable this functionality.

Reports Setup Sequence

In order to use the reporting tool, you will need to set up reporting options. Refer to Configuring The System To Enable Reports for more information.

XML Application Integration Setup Sequence

In order to use the XAI tool for sending information between third parties, you will need to set up XAI control tables. Refer to XML Application Integration for more information.

Case Management Setup Sequence

Case management options need only be set up if your organization uses cases to manage issues. Refer to Setting Up Case Types for more information.

Umbrella Agreement Management Setup Sequence

Umbrella agreement management options need only be set up if your organization uses umbrella agreements to manage contracts. Refer to the integration documentation for more information.

Outage Management Setup Sequence

Outage management options need only be set up if your organization interfaces with Oracle Utilities Network Management System. Refer to the integration documentation for more information.

Prepaid Metering Setup Sequence

Prepaid metering options need only be set up if your organization offers prepaid metering service to your customers. Refer to Defining Prepaid Metering Options for more information.

Zone Set Up

Most zones are delivered with the base-package and do not require any configuration. However, some zones are only available if configured by your implementation. Refer to Configuring Zones for more information.

To Do Options Setup

Refer to Setting Up To Do Options for more information on how to configure the system to match your organization's To Do management needs.

Meter Data Management Table Setup Sequence

This section provides the suggested order for the setup of Meter Data Management administrative data.

Sequence	Entity	Description
1	Unit of Measure	Quantities measured and recorded by the system (CCF, KWH, KW, etc.).
1	Service Quantity Identifier	Used to further distinguish between measured quantities that have identical UOM/TOU combinations.
1	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.).
1	Factor	Centrally stored sets of values for use in validation rules, bill determinants calculations, and other processes.
1	Market	Defines jurisdictions or regulatory environments in which a Service Point participates.
1	Measurement Cycle	Defines the schedule for manual meter reading of devices at Service Points in that cycle
1	Measurement Cycle Schedule	Define the dates on which devices are scheduled to be read for a given measurement cycle.
1	Outbound Message	Defines messages sent to external systems.
2	External System	Defines External Systems with which Oracle Utilities Meter Data Management should be able to communicate.
3	Head End Systems External Applications Market Participants	External entities that serve various roles relative to the application (head-end systems, billing systems, market participants, outage management systems, etc.).
3	Contact Type	Defines properties of a class of entities (businesses, persons).
4	Activity Type	Defines properties common to a specific type of activity.
4	Communication Type	Define properties common to a specific type of communication.

4	Service Task Type	Defines specific types of tasks performed by external users (self-service meter reads, self-service outage notifications, etc.)
4	Dynamic Option Type	Defines information common to dynamic options of a specific type.
4	Manufacturer	Individual companies that makes devices. Manufacturers also reference models.
4	Exception Type	Defines properties common to VEE Exceptions of a specific type.
5	VEE Group	Collections of VEE Rules that are applied to initial measurement data.
6	VEE Rule	Standard and custom VEE Rules that perform checking and/or manipulation of initial measurement data.
7	VEE Eligibility Criteria	Dictates whether a VEE Rule can execute based on a set of defined criteria.
8	Measuring Component Type	Defines the most important properties of a measuring component.
9	Device Configuration Type	Defines properties of Device Configurations of a given type.
10	Device Type	Defines information about a class of devices.
11	Service Point Type	Defines specific types of points at which service is delivered.
12	TOU Group	Used to limit the set of Time Of Uses that are usable in a TOU schedule.
13	TOU Map Template	Schedules used for TOU map data generation.
14	TOU Map Type	Define important properties of TOU maps of a given type.
15	Usage Transaction Exception Type	Defines properties common to Usage Transaction Exceptions of a specific type.
15	Usage Group	Collections of Usage Rules that are applied to measurement data to calculate bill determinants for Usage Subscriptions.
16	Usage Rule	Defines rules that perform calculations on measurement data to generate bill determinants and other values used by external systems.
17	Usage Rule Eligibility Criteria	Dictates whether a Usage Rule can execute based on a set of defined criteria.
18	Usage Subscription Type	Defines collections of properties common to a set of Usage Subscriptions.
19	Processing Methods (for Service Provider)	Control various behaviors for a service provider within the system such as which message is sent, how an external value is translated, among others.
19	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.
19	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.
19	Analytics	Defines the extract parameters, the bucket configurations and configuration snapshots used for extracting data for Oracle Utilities Analytics

Customer To Meter Batch Processing

This section provides guidelines for configuring batch processing with Oracle Utilities Customer To Meter.

Batch Process Dependencies

The contents of this section illustrate the periodicity and dependencies between the various background processes described above.

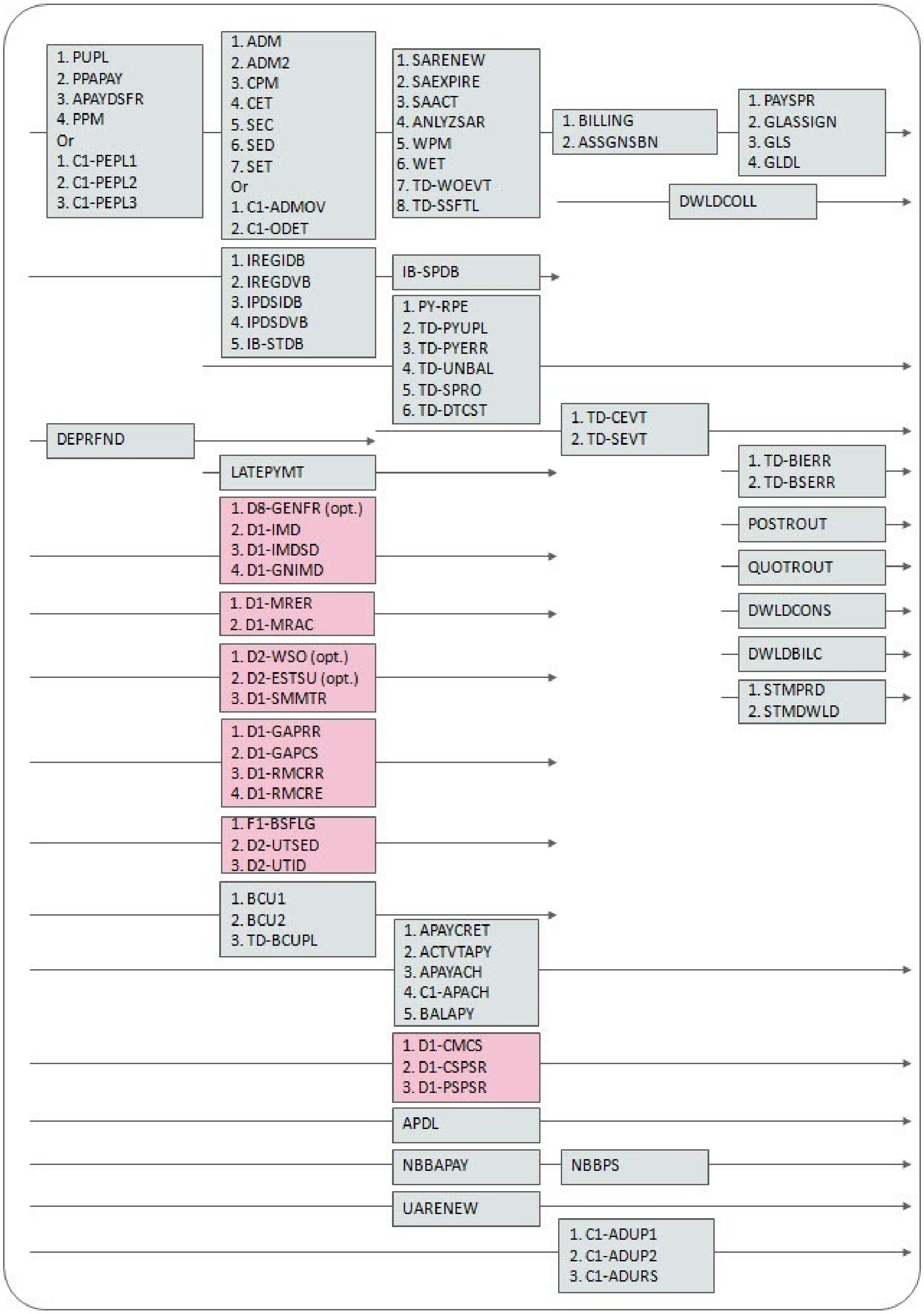
Batch Schedulers and Return Codes

If you use a batch scheduler (e.g., Control-M, Tivoli) to control the execution of your batch processes, it will be interested in the possible values of each process's return code. The return code is a number that indicates if the process ended successfully. All product processes will return one of the following return code values:

- 0 (zero). A value of zero means the batch process ended normally.
- 2. A value of 2 means the batch process detected a fatal error and aborted.

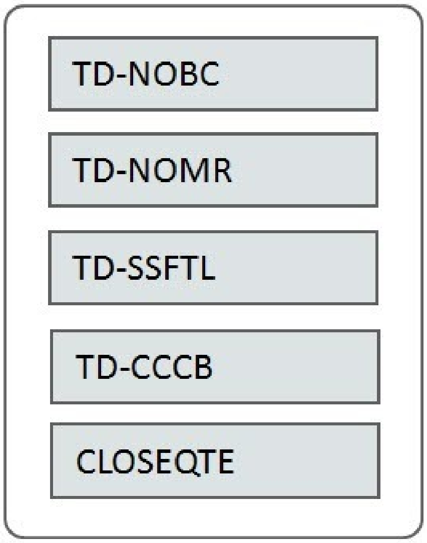
The Nightly Processes

The following diagram illustrates the dependencies between the batch processes.



The mnemonics in the boxes refer to the individual batch processes described above. When a box contains multiple processes, these processes must be run sequentially. When multiple boxes exist on a timeline, all processes in an earlier box must execute before the subsequent box is executed. Those timelines that appear beneath the Main Job Stream's timeline indicate when the timeline's respective processes can be executed in respect of the Main Job Stream.

The following diagram illustrates the daily batch processes for which there are no dependencies.

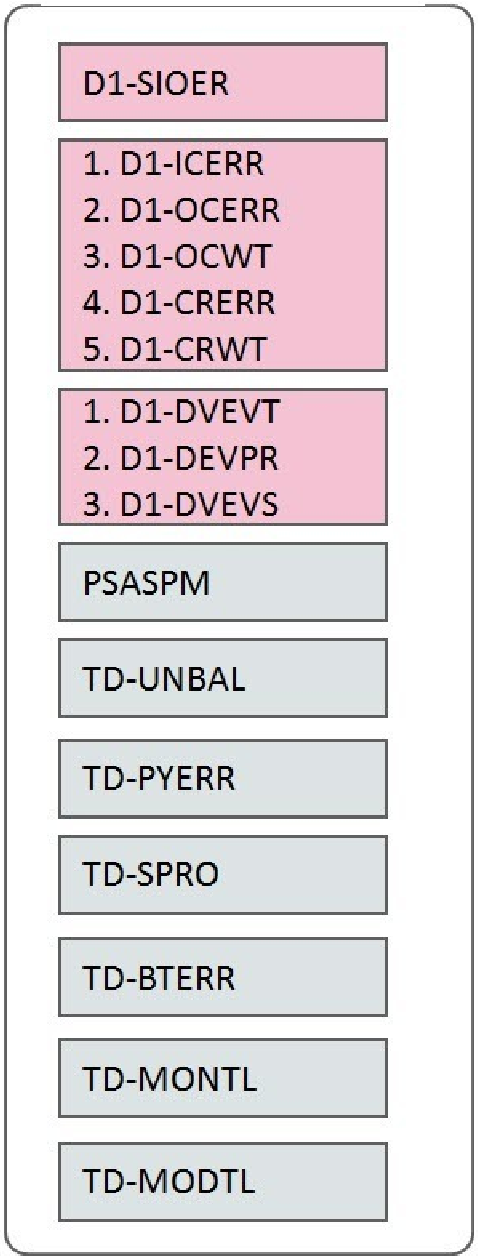


The mnemonics in the boxes refer to the individual batch processes described above.

NOTE:
No dependencies exist. As you can see, there are no dependencies between the boxes (meaning they may be run in parallel).

The Ongoing or Hourly Processes

The following diagram illustrates the dependencies between the ongoing or hourly batch processes.



The mnemonics in the boxes refer to the individual batch processes described above. When a box contains multiple processes, these processes must be run sequentially.

> **NOTE:**
> **No dependencies exist.** As you can see, there are no dependencies between the boxes (meaning they may be run in parallel).

Throughout the course of the day, you will likely want to run jobs to bring measurement and event data in from the various metering / head-end systems and other external systems. To accompany this, you should consider running the following batch processes:

- **Ongoing Master Data Sync Processing**

With the Oracle Utilities Customer to Meter deployment, data synchronization is used to keep certain objects synchronized. Refer to the C2MIntegration section for further details. Cleaning out synch staging records in error by running D1-SIOER may be necessary.

- **Command Processing**

  If using Oracle Utilities Smart Grid Gateway to process device commands, it’s important to keep communications flowing to and from smart meters to provide the most accurate picture of the state of a given meter. This would include:

  - Retrying inbound communications in error (D1-ICERR)
  - Retrying outbound communications in error (D1-OCERR)
  - Processing outbound communications waiting for a response (D1-OCWT) to see if they should be timed out.
  - Processing command request activities in error (D1-CRERR) and those that are waiting (D1-CRWT)

  ➤ **NOTE:** Base package business objects for communications and command activities are designed to trap any processing errors encountered and transition the object into an Error state. To deal with unexpected errors that can't be trapped, which could leave communications / command activities in unmonitored states, implementations can choose to configure their own batch controls based on the delivered D1-ACTVY and D1-COMM batch controls, restricting records processed by business object or maintenance object as needed

- **Initial Measurement Processing**

  For initial measurement processing, the following batch processes should be scheduled on an ongoing basis per business requirements:

  - Processing initial measurements in the pending state (D1-IMD).
  - Processing initial measurements seeders in error (D1-IMDSD).
  - Processing initial measurements in the exception state (D1-GNIMD).

  These processes are also highlighted in the Nightly Processes stream to reflect the need to bring in as much measurement data as possible prior to running billing, both up-to-date data from late-reporting meters and corrected readings from the head-end system.

- Event Processing

  The base package is configured such that device events are processed immediately upon receipt, since they might need to be sent to some other application such as an outage system. This can be changed by configuring a monitor process on the device event business object to stop records in a specified state, and then use a batch process to process the events all at once. Beyond this type of batch-oriented processing for events, other event processing could include:

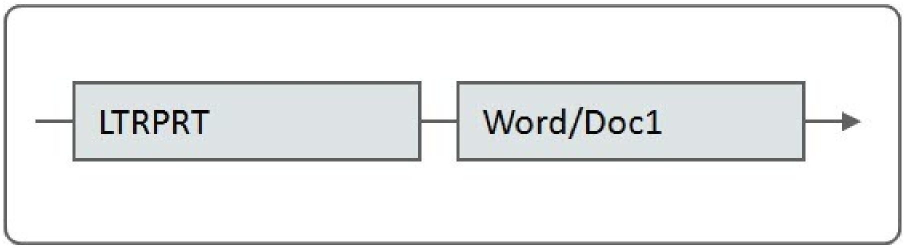
  - Re-processing device event "seeders" in error (D1-DVEVS)
  - Processing all device events that can come in as pairs (D1-DEVPR)
  - Picking up device events for processing if they've stopped in any state (D1-DVEVT)

  If device events are configured to be held from being sent onto downstream applications, such as to prevent "flicker" outage events (an outage event and a restore event received in rapid succession) from being sent, device event monitoring (D1-DEVPR and D1-DVEVT) should be set up to be run periodically to ensure timely transmission of events.

The Letter Processes

To extract information for your various letters, only one background process, LTRPRT, is required regardless of the different types of letters you have. This process simply calls an algorithm plugged-in on the respective letter template to construct its flat-file content.

The following diagram illustrates the dependencies for the letter background process. While this process should be run at least on a daily basis, you may want to consider running it more frequently (depending on how frequently you produce letters).



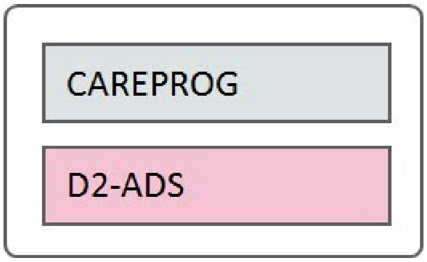
The mnemonics in the boxes refer to the individual batch processes described above. When a box contains multiple processes, these processes must be run sequentially. When multiple boxes exist on a timeline, all processes in an earlier box must execute before the subsequent box is executed.

The Periodic Processes

The following diagrams illustrate the dependencies between the periodic background processes. You may want to consider running these processes more or less frequently than suggested (depending on business requirements).

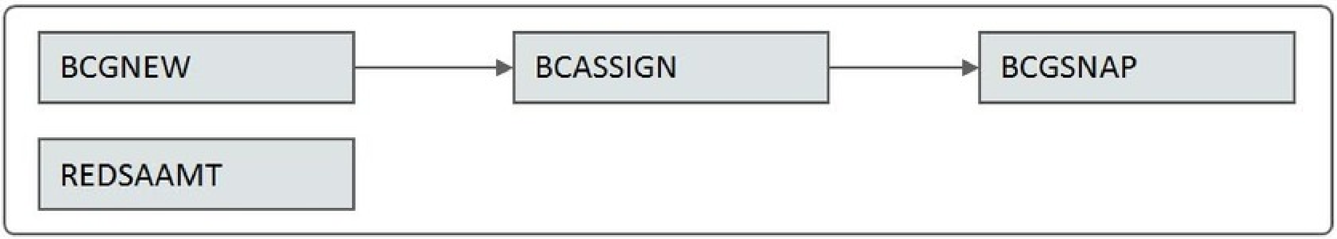
Periodic Processes: Weekly

The following processes should be run roughly once a week, or as per business requirements.



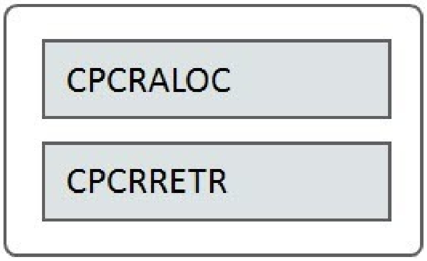
Periodic Processes: Monthly

The following processes should be run at least once a month, or as per business requirements.



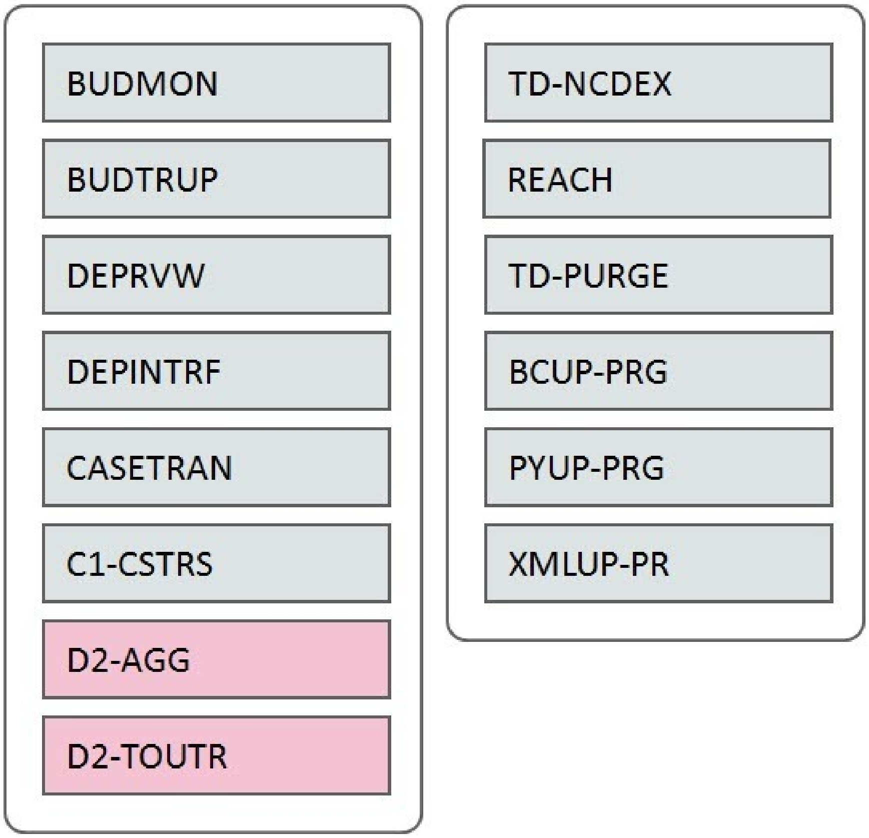
Periodic Processes: Annually

The following processes should typically be run once a year, or as per business requirements.



Periodic Processes: Per Business Requirements

The following processes should be run as frequently as dictated by business requirements.



The mnemonics in the boxes refer to the individual batch processes described above.

NOTE:
Few dependencies exist. As you can see, there are few dependencies between the boxes (meaning they may be run in parallel).

TOU Map Data Generation

TOU map data must be in place for all TOU maps used in usage calculations. Generation of this data is performed using the “Time of Use Map Data Generation Monitor” batch control (D1-TOUTR). This process can be performed for long time periods, such as a year, generating data for all time-of-use maps for the entire following year, or it could be done more frequently, such as whenever schedules are updated via the TOU map templates

Defining General Options

This section describes control tables that are used throughout Oracle Utilities Customer Care and Billing.

Defining Installation Options

The topics in this section describe the various installation options that control various aspects of the system that are specific to the Oracle Utilities Customer Care and Billing product.

- **FASTPATH:**
  Refer to Installation Options - Framework for options that are common to products on the same framework.

Installation Options - Main

Select **Admin** > **General** > **Installation Options** and use the **Main** page to define system-wide installation options.

Description of Page

Use **Quick Add Tender Type** to define the tender type defaulted on payments added using the Payment Quick Add transaction.

Use **Starting Balance Tender Type** to define the tender type of the starting balance recorded on your tender controls (this will almost always be the tender type associated with "cash"). This value is used during tender control balancing as a separate balance is required for each tender type in order to balance a tender control. Refer to The Lifecycle Of A Tender Control for more information.

- **FASTPATH:**
  For more information, refer to Setting Up Tender Types.

Turn on the **Create Field Activity Start Stop** if field activities should be created when a start or stop is recorded (as opposed to shortly before the start/stop date). You might want to turn this switch off if it's possible for the state of the service point (or its meter / item) to change between the time service is requested and the actual service date. Why? Because the state of the service point and the state of its meter / item affects the type of field activity that is created. For example, if a customer wants to start service and there is no meter at the metered service point, an "install meter" field activity is created. However, if by the time the install date comes around, a meter has been installed by some other means; this field activity is inappropriate. This is why you might want to setup the system to wait until shortly before the service date to create the field activity (i.e., it reduces the likelihood that an inappropriate field activity is created). Refer to Starting Service and Field Activities for more information.

If you use orders to create new customers, define the **Campaign** that should be defaulted on orders created when the order transaction is opened for a new customer. Refer to Real time Marketing of Services to a New Customer for more information.

Use the **Premise Geo Type** to indicate whether at least one geographic identifier (e.g., GPS coordinate) is Required or Optional on a premise. Refer to Defining Geographic Types for more information.

The **Alternate Representation** flag should be set to None unless your organization uses multiple character sets for a person's main name and / or a premise's address. Alternate representations are typically only used in countries where multiple character sets are used. For example,

- In Hong Kong, a person's name may be written in both Chinese characters and in English.
- In Japan, a person's name may be written in both Kanji and Katakana.

In both of the above situations, users need to be able to use both representations to find a customer or a premise.

➤ **NOTE:**
Spouses. If your organization doesn't use multiple character sets, you might want to consider using this functionality for spousal relationships. For example, rather than setup a person for each party in a spousal relationship, you could simply define one party using the person's main name and the spouse using the alternate name. While this is a bit of a "hack", it might be sufficient for your implementation as it will be much easier for an end user to use.

Alerts that should appear adjacent to a person's name or address. If your organization doesn't use multiple character sets, you might want to consider using this functionality to implement critical person or premise alerts. For example, if you have a customer who's supported by a specific account representative, you could enter the account rep's name as the person's alternate name. If you do this, the account rep's name would appear in parenthesis following the customer's name. In addition, you can search for the customers supported by the account rep on Control Central by entering the account rep's name. This is a bit of a "hack", but it might prove useful for a variety of functions.

If your organization uses alternate representations of person name or address, set this flag to one of the following values:

- Use a value of Address if you only use alternate representations for premise addresses.
- Use a value of Name if you only use alternate representations for a person's primary names.
- Use a value of Name & Address if you use alternate representations for both premise addresses and person names.

The following points describe the ramifications of this flag in the system:

- If you support alternate representations of a person's primary name,
- The name grid on Person - Main allows you to specify an Alternate name for the person.
- If you use the base package name formatting algorithm, a person's name will be shown throughout most of the system in the format AAA (BBB), where AAA is the person's primary name and BBB is the person's alternate name. Note, this format does not apply to names that appear in search results (i.e., the alternate name is not concatenated to the main name in search results; however you can search for information using the alternate name).
- Most of the system's person name-oriented searches will allow users to use both a person's primary and alternate names to search for information.
- If you support alternate representations of a premise's address,
- A new tab is available on the Premise page that allows a user to define an alternate address for a premise.
- If you use the base package address formatting algorithm, a premise's address will be shown throughout most of the system in the format AAA (BBB), where AAA is the premise's primary address and BBB is the premise's alternate address.

- Most of the system's premise-oriented searches will allow users to use both a premise's primary and alternate addresses to search for information.

If the CTI Integration flag has been enabled, set the **CTI Integration** flag to Yes if your organization integrates with an external computer telephony integration (CTI) system that supports a "get next caller in the queue" function. If this flag is set to Yes, then the **Next Call** button will appear in the application toolbar allowing customer service representatives to request the next customer waiting in the queue to speak to a CSR.

The **Control By CIS Division** flag affects system behavior in many areas. This flag should be set if your business operates in different jurisdictions and CIS Division is used to delineate accounts and premises that users can access. When this flag is set, all users must be associated with at least one CIS division. See Users and CIS Division for more information.

The following system behaviors are affected by this flag:

- On Person, CIS Division is presented and is required when adding an account. Customer class is defaulted after CIS Division is set.
- On Order, CIS Division is presented when a new account is being added and is required. Customer class is defaulted after CIS Division is set..
- On Person, Order, Account and Premise, the list of CIS Divisions that can be chosen is restricted to the CIS Divisions associated with the user. The user’s primary CIS Division is defaulted.
- On Account, CIS Division is presented before customer class. Customer class is defaulted after CIS Division is set.

The **Premise/SP Search** flag is used to provide access security to premises and service points that are not linked to an account.

- When the flag is set to Not Restricted, access to premises and service points that are not linked to an account is not secured.
- When the flag is set to Restricted by CIS Division, access to premises and service points that are not linked to an account is secured by the user’s CIS Divisions. If the premise has a CIS Division, then only users that have access to that CIS division can access that premise or its service points. See Users and CIS Division for more information. Premise and service points belonging to premises without a CIS Division. Are not secured. A Premise Type setting can control whether premises of that type require a CIS Division.

NOTE: Premises and service points that are linked to an account are secured by the account access group

WARNING:
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 this field's 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. Refer to Caching Overview **for information** on how to clear the system login cache (this is the cache in which installation options are stored).

Installation Options - Person

Select **Admin** > **General** > **Installation Options** and use the **Person** page to define person-specific installation options.

Description of Page

Use the **Person ID Usage** to indicate whether or not at least one form of identification is Required or Optional when a new person is added.

Each form of identification has an identifier type. For persons that are humans (as defined by the person type), the system defaults the identifier type defined in **Identifier Type (Person)**. For persons that are businesses (as defined by the person type), the system defaults the identifier type defined in **Identifier Type (Business)**.

Installation Options - Account

Select **Admin** > **General** > **Installation Options** and use the **Account** page to define account-specific installation options.

Description of Page

When a new account is added, the system requires it have a customer class. If the main customer linked to the account is a human (as defined by the customer's person type), the system defaults the customer class defined in **Customer Class (Person)**. For persons that are businesses (as defined by the person type), the system defaults the customer class defined in **Customer Class (Business)**. For more information, refer to Setting Up Customer Classes.

In addition to requiring a customer class when a new customer is added, the system also requires a "main customer" (i.e., a reference to a person who is identified as the main customer for the account). Enter the default **Account Relationship Type Code** to be used to define the main customer relationship. For more information, refer to Setting Up Account Relationship Codes.

Enter the default **Bill Route Type** to be used to define how bills should be routed to a customer. For more information, refer to Setting Up Bill Route Types.

Enter the default **Quote Route Type** to be used to define how quotes should be routed to a customer. For more information, refer to Setting Up Quote Route Types.

If the number of pending start and pending stop service agreements exceeds the **Start Stop Detail Threshold** for an account, it is considered a large account for start stop purposes. Refer to Start/Stop Maintenance for more information.

Installation Options - Billing

Select **Admin** > **General** > **Installation Options** and use the **Billing** page to define billing-specific installation options.

Description of Page

The **Bill Segment Freeze Option** controls when a service agreement's balance and the general ledger is affected by bill segments and certain types of adjustments. Refer to Preventing SA Balances And The GL From Being Impacted Until Bill Completion to understand the significance of this option.

The **Accounting Date Freeze Option** controls how the accounting date defined on financial transactions is populated. Refer to Forcing The Freeze Date To Be Used As The Accounting Date to understand the significance of this option.

Define the **Rollover Threshold Factor** used by billing to determine if a register's consumption is sensible. This value is used as follows:

- Whenever billing calculates a meter's register's consumption, it compares it to a value equal to X times the register's maximum capacity (where X is the Rollover Threshold Factor).
- If consumption exceeds this value, a bill segment error is generated. If this consumptive value is correct, a user will need to override the consumption value billed on the bill segment (billing will never use such a read).

Define the **Minimum Amount for Final Bill**. If a final bill is less than this amount, the bill is still produced; it's just not printed.

Typically, the system sets a bill's Bill Date equal to the date on which it is completed. If you want to be able to specify a bill's Bill Date when you complete a bill, turn on **User Can Override Bill Date**. You would only want to override the bill date if you are setting up sample bills from historical period whose bill date needs to reflect the respective historical period.

Turn on **Use High Low Failures on Bill** if the system should mark meter reads that fail high / low checks as billable. Turn off this switch if such reads should not be used by billing. Users may override this default value on a specific read. Refer to Validations for more information.

Base Time is used by billing algorithms to determine the effective start and end times for a given period. The **Start Day Option** further defines how to use the base time, indicating whether the base time is for the Current Day or for the Previous Day.

Turn on **Use Alternative Bill ID** if your implementation uses assigned document numbers or sequential bill numbers. In the **Alternative Bill ID Option** list:

- Select **Document Numbers** if you require a system-assigned document number for each bill in addition to the Bill Id, which is a system-assigned random number used as the bill's primary identifier. Refer to Document Numbers for more information.
- Select **Sequential Bill Numbers** if you require a system-assigned unique sequential number for each bill in addition to the Bill Id, which is a system-assigned random number used as the bill's primary identifier. Refer to Sequential Bill Numbers for more information

➤ **NOTE:**
Document Number Algorithms. In addition to turning on **Use Alternative Bill ID** and specifying the **Alternative Bill ID Option**, the Document Number and Document Number Details algorithms must be enabled on the Installation record. These algorithms contain the logic used by the system to assign a document number to a bill.

The **Bill Correction** option lets you control whether your implementation uses Credit Notes or Correction Notes. Select the **Credit Note** option if you require bill segment cancellation details to be presented to the customer on a separate bill (referred to as a credit note). Refer to Credit Notes for more information. Select the **Correction Note** option if you require bill segment cancellation details and bill segment rebill details to be presented to the customer on a separate bill (referred to as a correction note). Refer to Correction Notes for more information.

➤ **NOTE:**
Credit Notes or Correction Notes. The Bill Correction option on the Installation record controls whether Credit Notes or Correction Notes are allowed. If your implementation uses Correction Notes, the override label on the following should be customized accordingly:

Lookup value CRNT on the customizable lookup field TXN_FLTR_TYPE_FLG (this lookup value is used on the Match Event Page and Account Bill History transactions)

Lookup value CR on the customizable lookup field PYCAN_SYS_DFLT_FLG (this lookup value is used on the Pay Cancel Reason transaction)

Metadata field CR_NOTE_FR_BILL_ID (this field is used on the Bill Search Page)

The **Autopay Creation Option** controls when automatic payments are created, distributed, and frozen. This option allows you to control when automatic payments will affect customer's balances and when their financial impact affects the general ledger. Refer to How And When Are Automatic Payments Created to understand the significance of this option.

Installation Options - C&C

Select **Admin** > **General** > **Installation Options** and use the **C&C** page to define credit and collections-specific installation options.

Description of Page

When you look at an account or service agreement's debt, the system shows the respective age of each piece of outstanding debt. The **Oldest Bucket Age (Days)** defines the debt age after which the system groups all outstanding debt together. For example, if this field is 180:

- The exact age of each element of debt that is less than 180 days old would be shown as a separate line item in the aged debt information.
- All debt older than 180 days would be amalgamated into a single "bucket".

Oldest Bucket Age (Days) also has another use - it defines the age of financial transactions that are considered by the background process that marks old debt as "redundant". This batch process is referred to by the batch code of REDSAAMT.

> **WARNING:**
> If you change the value of **Oldest Bucket Age (Days)**after debt has been marked as "redundant" by REDSAAMT, the system will NOT re-age the old debt (i.e., once a financial transaction has been marked as "redundant", it is "redundant" forever).

Enter what you consider to be an excellent credit rating in **Beginning Credit Rating**. Collection events can cause an account's credit rating to decrease. When an account's credit rating falls below a certain level, different collection processes may ensue.

Use **Beginning Cash-Only Score** to define the cash-only score for accounts with a perfect payment history (i.e., one without non-sufficient funds). When you cancel a payment tender and use a cancellation reason marked as NSF, the system will cause the account's cash-only score to increase by the value on the payment cancellation reason.

Use **Credit Rating Threshold** to define when an account's credit rating becomes risky. When an account's credit rating falls beneath the Credit Rating Threshold, the system will:

- Assuming you've enabled the Control Central alert algorithm, C1-CRRT-ACCT, an alert displays when an account's credit rating falls below the credit rating threshold on the installation record. This algorithm is plugged-in on the installation record.
- Subject the account's debt to different collection criteria. For more information, refer to Designing Your Collection Class Control Overrides.

Use **Cash-Only Threshold** to define the number of cash-only points a customer must have before the system warns the CSR accepting payments that the account is cash-only.

Installation Options - Financial Transaction

Select **Admin** > **General** > **Installation Options** and use the **Financial Transaction** page to define financial transaction installation options.

Description of Page

Use **G/L Batch Code** to define the batch process that is used to interface your financial transactions to your general ledger. The process is snapped on FT download records by the GLS background process.

Use **A/P Batch Code** to define the batch process that is used to interface your check requests (initiated with adjustments with an adjustment type that reference an A/P request type) to you accounts payable system.

Use **Fund Accounting** to indicate if fund accounting is Practiced or Not Practiced at your organization.

Use **Alternate Currency** to indicate if your organization accepts customer payments in currencies other than the account's currency.

- **FASTPATH:**
  Refer to Alternate Currency Payments to understand the significance of this option.

Installation Options Framework - Algorithms

Select **Admin** > **General** > **Installation Options Framework** and use the **Algorithms** page to define specific system events.

The following table describes each **System Event**.

System Event	Optional / Required	Description
Account Information	Optional	We use the term "Account information" to describe the basic information that appears throughout the system to describe an account. The data that appears in "account information" is constructed using this algorithm. Plug an algorithm into this spot to override the system default "Account information". Click here to see the algorithm types available for this system event.
Adjustment Information	Optional	We use the term "Adjustment information" to describe the basic information that appears throughout the system to describe an adjustment. The data that appears in "Adjustment information" is constructed using this algorithm. Plug an algorithm into this spot to override the system default "Adjustment information". Note: This algorithm may be further overridden by an "Adjustment information" plug-in on the Adjustment Type. Refer to Adjustment Type for how algorithms of this type are used. Click here to see the algorithm types available for this system event.
Appointment Information	Required	We use the term "Appointment information" to describe the basic information that appears throughout the system to describe an appointment. The data that appears in "appointment information" is constructed using this algorithm. Click here to see the algorithm types available for this system event.
Automatic Payment Creation	Required if you allow customers to pay automatically	This algorithm is executed to create automatic payments whenever the system creates automatic payments. Refer to How And When

		Are Automatic Payments Created for the details. Click here to see the algorithm types available for this system event.
Bill Information	Required	We use the term "Bill information" to describe the basic information that appears throughout the system to describe a bill. The data that appears in "bill information" is constructed using this algorithm. Click here to see the algorithm types available for this system event.
Bill Segment Information	Optional	We use the term "Bill segment information" to describe the basic information that appears throughout the system to describe a bill segment. The data that appears in "bill segment information" is constructed using this algorithm. Click here to see the algorithm types available for this system event.
Case Information	Optional	We use the term "Case information" to describe the basic information that appears throughout the system to describe a case. The data that appears in "case information" is constructed using this algorithm. Plug an algorithm into this spot to override the system default "Case information". Note: This algorithm may be further overridden by a "Case information" plug-in on the Case Type. Refer to Case Type for how algorithms of this type are used. Click here to see the algorithm types available for this system event.
Collection Agency Referral Information	Optional	We use the term "Collection Agency Referral information" to describe the basic information that appears throughout the system to describe a collection agency referral. Plug an algorithm into this spot to override the system default "collection agency referral information". Click here to see the algorithm types available for this system event.
Collection Process Additional Information	Optional	This algorithm displays additional information related to a collection process in a special field on the collection process main page. Click here to see the algorithm types available for this system event.
Control Central Alert	Optional	There are two types of alerts that appear in the Alert Zone and on Payment Event - Main: 1) hard-coded system alerts and 2)

		alerts constructed by plug-in algorithms. You cannot change the hard-coded alerts (see the Alert Zone for the complete list). However, by plugging in this type of algorithm you can introduce additional alerts. An error displays if more than 60 alerts are generated for an account by plug-in algorithms. Click here to see the algorithm types available for this system event.
Credit Rating "Created By" Information	Required	The data that appears in the credit rating "created by" information is constructed using this algorithm. Refer to Account - C&C for more information about the credit rating. Click here to see the algorithm types available for this system event.
Credit Rating History Information	Optional	We use the term Credit Rating History information to describe the basic information that appears throughout the system to describe a credit rating history entry. Plug an algorithm into this spot to override the system default "credit rating history information". Click here to see the algorithm types available for this system event.
Document Number	Optional	If document numbers have been enabled on the installation record, this algorithm type assigns a document number to a bill or payment event. Click here to see the algorithm types available for this system event.
Document Number Details	Optional	If document numbers have been enabled on the installation record, this algorithm type is responsible for returning the details used to construct the document number. Click here to see the algorithm types available for this system event.
Determine Open Item Bill Amounts	Required if you use overdue functionality to collect on bills	This algorithm is responsible for determining the unpaid amount of an open-item bill. It can also be used to return the unpaid amount for a specific SA on a bill. Click here to see the algorithm types available for this system event.
FA Additional Information	Optional	This algorithm displays additional information related to a field activity in a special field called Additional Info on the field activity main page.

		For example, contact information linked to the field activity's field order may be displayed. Click here to see the algorithm types available for this system event.
FA Information	Required	We use the term FA information to describe the basic information that appears throughout the system to describe a field activity. The data that appears in "FA information" is constructed using this algorithm. Click here to see the algorithm types available for this system event.
Item Information	Required if you have items	We use the term "Item info" to describe the basic information that appears throughout the system to describe an item. The data that appears in "Item info" is constructed using this algorithm. Click here to see the algorithm types available for this system event.
Meter Information	Required if you have meters	We use the term "Meter info" to describe the basic information that appears throughout the system to describe a meter. The data that appears in "Meter info" is constructed using this algorithm. Click here to see the algorithm types available for this system event.
Meter Read High Low Limits	Not used with Customer To Meter	Click here to see the algorithm types available for this system event.
Online Bill Display	Optional	This algorithm constructs a PDF that contains the image of a bill. This algorithm is executed when the Display Bill button is clicked on the Bill page. Refer to Technical Implementation of Online Bill Image for more information. Click here to see the algorithm types available for this system event.
Online Field Order Image	Optional	This algorithm constructs a PDF that contains the image of a field order. Click here to see the algorithm types available for this system event.
Online Letter Image	Optional	This algorithm constructs a PDF that contains the image of a letter. This algorithm is executed when the Display Letter button is pressed on Customer Contact - Main. Refer to Technical Implementation of Online Letter Image for more information. Click here to see the algorithm types available for this system event.
Online Quote Image	Optional	This algorithm constructs a PDF that contains the image of a quote. This algorithm is executed when the Display Quote button is

		pressed on Quote - Main. Refer to Technical Implementation of Online Quote Image for more information. Click here to see the algorithm types available for this system event.
Online Statement Image	Optional	This algorithm constructs a PDF that contains the image of a statement. This algorithm is executed when the Display Statement button is pressed on Statement - Main. Refer to Technical Implementation of Online Statement Image for more information. Click here to see the algorithm types available for this system event.
Override Proration Factors	Optional	This algorithm is only used if your organization has unusual rate proration requirements that necessitate the overriding of the base package proration logic. For example, you may have certain calculation rules whose charges should never be prorated. Refer to Overriding Proration Factors for more information. Click here to see the algorithm types available for this system event.
Override Seasonal Proration	Optional	This algorithm is only used if your organization has unusual method of determining the seasons for your calculation rules. For example, you may determine the seasonal boundaries for a calculation rule based on the scheduled meter read date associated with the bill cycle. Click here to see the algorithm types available for this system event.
Payment Amount Calculation	Required	This algorithm is executed to calculate the amount of an automatic payment for a bill for an account with an active auto pay option. Refer to How And When Are Automatic Payments Created for more information on automatic payments. This algorithm is also executed to default the amount of a manually added payment. Refer to How To Add A New Payment Event for more information on adding a payment manually. Click here to see the algorithm types available for this system event.
Payment Information	Required	We use the term "payment information" to describe the basic information that appears throughout the system to describe a payment. The data that appears in "payment information" is constructed using this algorithm.

		Click here to see the algorithm types available for this system event.
Person Information	Required	In most parts of the system, a person's Main name is displayed to describe a person. However, several transactions do not use this method. Rather, these transactions call the algorithm that's plugged into this spot to construct the person's name. Refer to the description of the **Alternate Representation** flag on the **Main** tab for a list of these transactions and for the rationale behind this algorithm. Click here to see the algorithm types available for this system event.
Person Name Validation	Required	The format of names entered on Person - Main and Order - Main is validated using this algorithm. Click here to see the algorithm types available for this system event.
Premise Information	Required	We use the term "premise info" to describe the basic information that appears throughout the system to describe a premise. The data that appears in "premise info" is constructed using this algorithm. Click here to see the algorithm types available for this system event.
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 on-line. This algorithm is used by the two on-line reporting pages to properly invoke the reporting tool from within the system. Click here to see the algorithm types available for this system event.
SA Information	Optional	We use the term "SA information" to describe the basic information that appears throughout the system to describe a service agreement. The data that appears in "SA information" is constructed using this algorithm. Plug an algorithm into this spot to override the system default "SA information". Note: This algorithm may be further overridden by an "SA information" plug-in on the SA Type. Refer to SA Type - Algorithms for how algorithms of this type are used. Click here to see the algorithm types available for this system event.

Severance Process Cancellation	Optional	This algorithm is executed to perform additional processing when the system cancels a severance process. Note: This algorithm is executed before the Severance Process Template - Post Cancel Algorithm is executed. Canceling a severance process on-line manually does not execute this algorithm. Click here to see the algorithm types available for this system event.
SP Information	Required	We use the term "SP info" to describe the basic information that appears throughout the system to describe a service point. The data that appears in "SP info" is constructed using this algorithm. Click here to see the algorithm types available for this system event.

Defining Customer Languages

As described under Defining Languages, you define the language in which each user see the system. In addition to defining each user's language, the system allows you to define each customer's preferred language. For example, one customer can receive bills in English whereas another customer could receive their bills in Chinese.

Each customer's language is defined by the language code on their person record. Bills, adjustments and other system-generated records will then be done in the language of the main customer of the account. In addition, the language code is passed on to all customer-facing interfaces, such as letter requests and bill print.

NOTE:
You can define Rates in multiple languages - when a bill is generated, the line-item descriptions are generated and stored in the account's main customer's language of choice. Anyone who subsequently views these bills can only see the descriptions in that language.

To support bills and other correspondence, you must also provide translations of standard bill stock and letters. This must be handled by your printing software vendor.

Defining Accounting Calendars

Accounting calendars determine the accounting period to which a financial transaction will be booked. The following points describe how the system determines a financial transaction's account period:

- Every financial transaction references an accounting date and its service agreement
- Every service agreement references a service agreement type
- Every service agreement type references a GL division

- Every GL division references an accounting calendar
- The accounting calendar contains the cross-reference between the accounting date specified on the financial transaction and related accounting period in your general ledger

WARNING:
This information must be the same as the information in your financial database.

To add or review an accounting calendar, choose **Admin** > **General** > **Accounting Calendar** > **Search.**

Description of Page

Enter a unique **Calendar ID** and **Description** for the calendar.

Enter the **Number Of Periods** for the calendar. Don't count the adjustment period, if you use one, or any special "system" periods.

Specify the **Fiscal Year**, each **AccountingPeriod** in that year, a **Period Description**, the **Begin Date** and the **End Date**.

When you enter begin and end dates, you can define monthly calendar periods or any fiscal period that matches your accounting calendar (weekly, bimonthly) as long as the begin and end dates of successive periods do not overlap. Every day of the year must be included in a period; do not leave gaps between period dates.

For each fiscal period, enter the **Open From Date** and **Open To Date**. These dates define when that particular business dates are open for posting financial transactions to that fiscal period. For example, you might calculate a bill on Sept 1 for usage recorded on 31 August. To post this financial transaction in the August period, you must keep it open through Sept 1.

As time passes, you will need to return to this transaction to manually enter ensuing years. You can enter several years at a time or incorporate the task into end-of-year system maintenance.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference CI_CAL_GL

Defining General Ledger Divisions

There are two types of Divisions referenced in the system: a CIS Division and a GL Division. This is a rather powerful structure, but it can be confusing.

- General Ledger divisions typically comprise individual entities (e.g., companies) in your general ledger. You must set up a GL division for each such entity. The GL division's sole purpose in the system is to define the accounting period associated with financial transactions linked to service agreements associated with the GL division (service agreements are associated with GL divisions via their SA type). The system cares about accounting periods in order to prevent a user from booking moneys to closed periods. It also uses accounting periods when it produces the flat file that contains the consolidated journal entry that is interfaced to your general ledger (refer to The GL Interface for more information).

  NOTE:
  When determining how many GL Divisions you need, be sure to consider your general ledger and how your chart-of-accounts is structured. You will typically have one GL division for each "company" in your general ledger.

- A CIS division is typically associated with a jurisdiction. The definition of a jurisdiction is a geographic-oriented entity with unique business rules. For example, if you conduct business in California and Nevada, and each state has different collection rules, you will need a separate jurisdiction for each state. You must set up a CIS division for each jurisdiction in which you conduct business.

FASTPATH:
Refer to CIS Division Portal for information about CIS Divisions.

To define a general ledger division, select **Admin** > **General** > **General Ledger Division.**

Description of Page

Enter a unique **GL Division** for the general ledger division.

Enter a **Description** of this general ledger division.

Define the accounting **Calendar ID** that controls how to convert an FT's accounting date into an accounting period. Refer to Defining Accounting Calendars for more information.

You may define a **Currency Code** for the GL division. Note that the system does not use this currency code.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference CI_GL_DIVISION.

Defining Banks & Bank Accounts

The topics in this section describe how to maintain your implementation's bank accounts.

Bank - Main

To review Banks choose **Admin** > **Financial** > **Bank** > **Search.**

Description of Page

Enter a unique **Bank Code** and **Description** for the bank.

The **Bank Accounts** collection displays the bank accounts currently linked to this bank code. Use the drill down button to view more details or to modify the bank account details. Alternatively, you may navigate to the Bank Account tab and scroll to the desired bank account.

Bank - Bank Account

To review Bank Accounts for a Bank, choose **Admin** > **Financial** > **Bank** > **Search** and then navigate to the **Bank Account** page.

Description of Page

Use the **Bank Accounts** tab to define the attributes of each bank account. For each account, enter the following information:

- Enter a **Bank AccountKey** to identify an Account at a Bank. You may have more than one account at a given bank, and you may have accounts at more than one bank. This code will allow the system to easily identify a specific account.
- Enter a **Description** to appear on prompt lists, inquiries, and reports.
- Enter the **Account Number**, **Check Digit** and if needed, the **Branch ID** of the bank where the account is held.
- Enter the **CurrencyCode** for the currency in which the account is denominated.

- Use **DFI ID** to define the Depository Financial Institution ID that is interfaced to the automatic payment-processing agent as part of the automatic payment interface.
- Enter the **Distribution Code** to be used for cash GL distributions when a payment is frozen or canceled.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference CI_BANK_ACCOUNT.

Defining Issuing Centers

This section provides information about defining issuing centers that are used to assign document numbers to bills and payment events. An issuing center should be configured for each location that issues bills. The installation record Document Number and Document Number Details algorithms contain the logic used by the system to assign a document number to a bill. To set up an issuing center, open **Admin** > **General** > **Issuing Center** > **Add.**

●

FASTPATH:
Refer to Document Numbers for information about document number assignment.

➤

NOTE:
This section is only relevant for some organizations. The information in this section is only relevant if your organization indicated on the installation record that it uses document numbers as an alternative bill id. If your organization does not use document numbers as an alternative bill id, then no other setup is required.

The topics in this section describe the base-package zones that appear on the Issuing Center portal.

Actions

This is a standard actions zone.

If the issuing center is in a state that has valid next states, buttons to transition to each appropriate next state are displayed.

Issuing Center List

The Issuing Center List zone lists every issuing center. The following functions are available:

- Click a broadcast button to open other zones that contain more information about the adjacent issuing center.
- Click the Add link in the zone's title bar to add a new issuing center.

Issuing Center

The Issuing Center zone contains display-only information about an issuing center, including its current and historic branches. This zone appears when an issuing center has been broadcast from Issuing Center List zone or if this portal is opened via a drill down from another page.

Please see the zone's help text for information about this zone's fields.

Issuing Center Log

This is a standard log zone.

Defining Service Tasks Options

This section describes concepts and procedures related to service tasks.

About Service Tasks

Service tasks are records that can be used to perform a variety of tasks. Examples include:

- Task-related records performed by users of other Oracle Utilities applications, such as Oracle Utilities Customer Self Service.
- Task-related records performed by Oracle Utilities Customer Care and Billing to manage specific processing, such as net energy metering true-ups, or prepaid billing to create bill segments for smart meter prepaid service agreements.

The main attribute used to define a service task is service task type. The service task type defines properties common to specific types of service tasks.

For more information, see Searching and Viewing Service Tasks.

About Service Task Types

Service task types define properties common to specific types of service tasks. Service task types represent different types of tasks that can be performed by:

- Users of other Oracle Utilities applications, such as Oracle Utilities Customer Self Service. An example of a service task includes self service meter reads, in which users enter their own meter reads via the Customer Self Service application.
- Oracle Utilities Customer Care and Billing to manage specific processing. Examples include net energy metering true-ups, or prepaid billing to create bill segments for smart meter prepaid service agreement.

Service task types can be defined by the following attributes:

- **Service Task Type**: the name of the task type.
- **Service Task Type Business Object:** the business object that defines the behavior of the service task type.
- **Service Task Business Object:** the business object instantiated when service tasks of this type are created.
- **Service Task Class:** the category used to define service task types for reporting purposes (self-service, miscellaneous, etc).
- Other data based on the specific type of service task (such as minimum days in true-up period, billing frequency, or default payment method.)

For more information about defining service task types, see Defining Service Task Types.

For more information about the service task types delivered with the application, see Base Package Service Task Types.

Defining Service Task Types

To maintain existing service task event types, select **Admin** > **Customer** > **Service Task Type,** then use standard actions to edit, duplicate, or delete a service task type.

To define a new service task type, follow these steps:

1. Select **Admin** > **Customer** > **Service Task Type.**.
2. Enter a name and a meaningful description for the service task type.
3. If needed, select the business object to use when creating service tasks of this type.
4. Select the service task class applicable to service tasks of this type, if applicable.

   Some service tasks types used solely by Oracle Utilities Customer Care and Billing have a default class of **Miscellaneous**.
5. Enter a detailed description for the service task type.
6. Complete the remaining fields and sections, as needed.
7. If applicable, select a To Do type and corresponding To Do role to use when creating To Do entries related to service tasks of this type.
8. Click **Save**.

This service task type can now be used when service tasks are received from other Oracle Utilities applications, such as Oracle Utilities Customer Self Service, or service tasks created specifically by Oracle Utilities Customer Care and Billing. For more information about service task types, see About Service Task Types.

For more information about the service task types delivered with the application, see Base Package Service Task Types.

Base Package Service Task Types

This section provides descriptions of the service task types provided with the base package.

Service Task Type	Business Object	Detailed Description	Related Transaction Business Object
Appointment Notification Task Type	C1-NotifyAppointmentTaskType	This business object is used to capture the information to use in appointment notification processing.	Appointment Notification Task
Create Customer Contact Task Type	C1-CreateCustContactTaskType	This business object is used to capture the information to use in create customer contact task processing.	Customer Contact Request Task
Direct Debit Task Type	C1-DirectDebitMandateTaskType	This business object defines configuration information that is used for processing Single Euro Payments Area (SEPA) direct debit transactions.	Direct Debit Mandate Task

FA Completion Task Type	C1-FACompletionTaskType	This business object captures the attributes used in field activity completion task processing.	FA Completion Task
Field Activity Remark Task Type	C1-FieldActivityRemarkTaskType	This business object is used to capture the information to use in create field activity remark task processing.	Field Activity Remark Task
Form Task Type	C1-FormTaskType	This business object captures attributes relevant for a particular form type to be used when a self-service customer creates a form.	Form Task
Missed Appointment Notification Task Type	C1-NotifyMissedApptTaskType	This business object is used to capture the information to use in missed appointment processing.	Missed Appointment Notification Task
Notification Task Type	C1-NotifyTaskType	This business object captures information about a notification that is made available for sign-up by self-service users.	Notification Preferences
Net Energy Metering Task Type	C1-NEMTrueUpTaskType	This business object is used to maintain the various configuration options that are used by the true up monitor (TUM) business object's algorithms. It defines the length of the true up period as well as the adjustments types used during the true up process. For true up reversals, the adjustment cancel reason to use is also captured here.	True Up Monitor Task
Outage Call Type	C1-OutageCallType	This business object defines the behavior of an outage call type used to support trouble calls captured in CCB and integrated to NMS.	Outage Call
Payment Arrangement Task Type	C1-PATaskType	This business object defines the expected behavior for when a self-service user requests a payment arrangement.	Payment Arrangement Task
PPB Payment Notification Task Type	C1-PPBPaymentNotifyTaskType	This business object captures information about the Prepaid Billing Payment Request notification that is made available for sign-up by self-service users.	Notification Preferences
Prepay Biller Task Type	C1-PrepayBillerTaskType	This business object defines the behavior of a prepay biller task type used in smart meter prepay billing.	Prepay Biller Task
Product Offer Publish	C1-ProductOffer	This business object contains the product offer elements that are relevant when publishing product	Not defined

		offers to Siebel Energy. It is used to read product offer information when building the initial and final snapshots for the sync request.	
Product Offer Task Type	C1-ProductOfferTaskType	This business object is used to define the behavior of a product offer. Product offers are configured in CCB and published to Siebel Energy.	Not defined
SA Creation Rule Publish	C1-SACreationRule	This business object contains the SA creation rule elements that are relevant when publishing product offers to Siebel Energy. It is used to read SA creation rule information when building the initial and final snapshots for the sync request.	Not defined
SA Creation Rule Task Type	C1-SACreationRuleTaskType	This business object defines the behavior of the SA creation rules for a product offer. Each SA creation rule indicates the type of service agreement to create for the product offer.	Not defined
Service Request Task Type	C1-ServiceRequestTaskType	This business object is used as a parent service task type business object for service request related task types such as appointment notification, missed appointment and customer contact creation requests.	Not defined
Siebel Customer Maintenance Task Type	C1-CustomerMaintenanceTaskType	This business object is used in the Siebel integration. When a customer maintenance request is received from Siebel, an inbound service determines the service task type to use for the event.	Siebel Customer Maintenance Task
Start/Stop Task Type	C1-StartStopTaskType	This business object defines the expected behavior for when a self-service user requests to start, stop or transfer service.	Start/Stop Request Task

System-wide Options

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 Data Management / Smart Grid Gateway 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.

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 dashboard 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: Associates a To Do entry via characteristic 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.

> **NOTE:**
> Additional detail on how the To Dos are associated to related data can be found in the detailed description of the D1-TDPRCRE algorithm 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

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.

➤ **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.

> **NOTE:**
> Refer to Configuring Multi-Time Zone Support for additional information on setting up multi-time zone support.

Time Zones

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

> **NOTE:** The server time zone must be correctly specified on the installation options record for the system to work properly.

For additional information see Defining in the Oracle Utilities Application Framework *Administrative User Guide*.

Configuring Multi-time Zone Support

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 Financial Transaction Options

Bills, payments and adjustments share one very important trait - they affect how much your customers owe. This section explains the financial design of the system and describes how to set up the tables that control the financial impact of these transactions.

➤ **NOTE:**
The tables in this section are the first of many that must be set up before you can create bills and apply payments. In this section, we limit the discussion to those tables that control the financial impact of bills, payments and adjustments. In later sections, we describe the tables that control other billing-related functions like meter reading and rates. It is only after all of these tables are set up that you will be able to generate the various financial transactions.

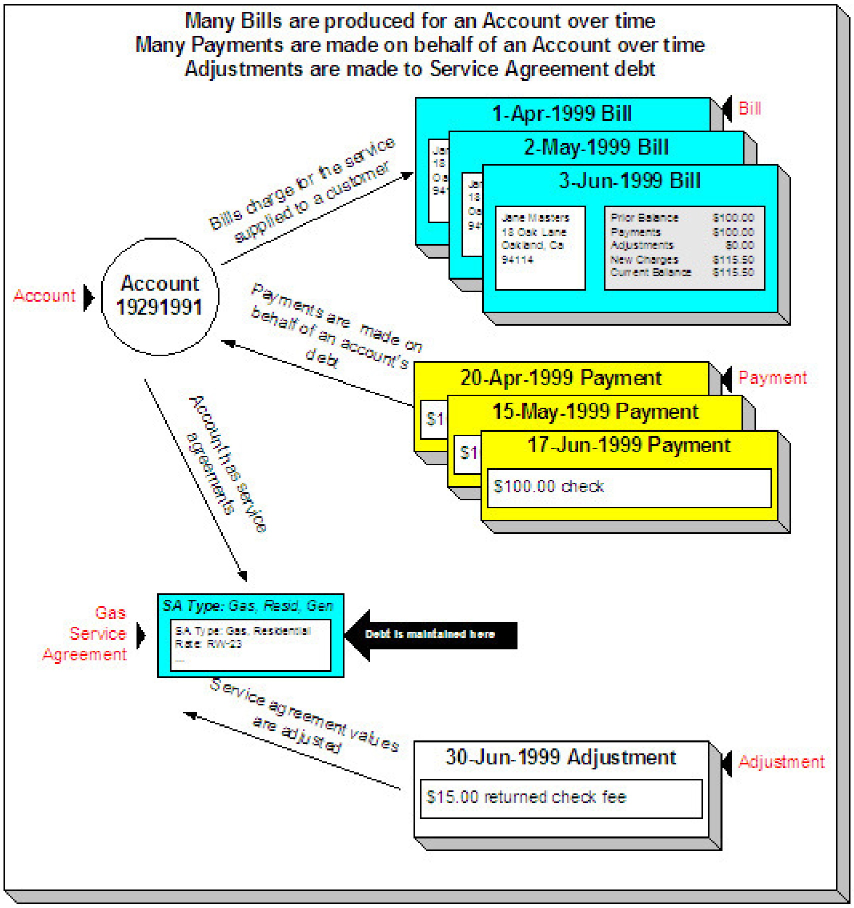
The Financial Big Picture

This section provides an overview of the relationship between an account and the various financial transactions that influence how much a customer owes.

WARNING:
If your organization practices cash accounting for payables (i.e., you only pay the taxing authority when you get paid), refer to Payables Cash Accounting. If you organization practices open-item accounting (i.e., payments must be matched to bills), refer to Open Item Accounting.

Bills, Payments & Adjustments

The following diagram illustrates the relationship between an account and its financial transactions:



The following concepts are illustrated above:

Bills are produced for accounts Over time, many bills may be produced for an account. For more information about a bill, see Bill Details.

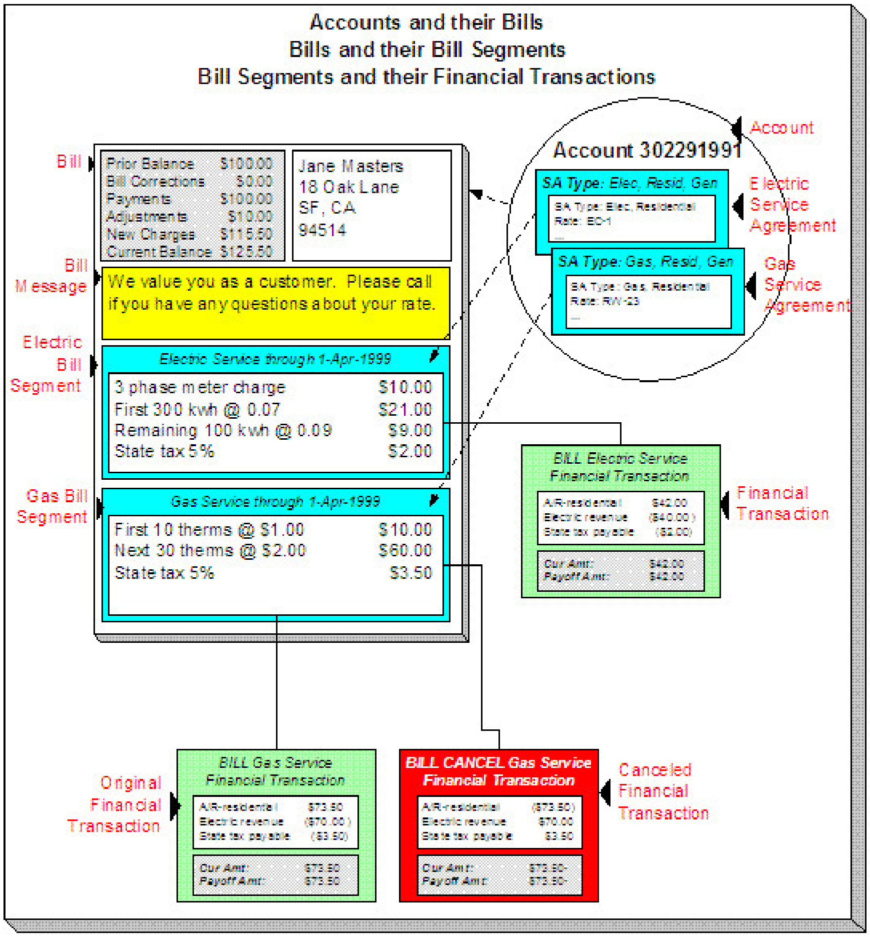
Payments are made for accounts Over time, many payments may be applied to an account's debt. For more information about a payment, see Payment Details.

Service agreements have debt The system maintains debt on each individual service agreement. An account's debt is the sum of its service agreements' debt.

Service agreements are adjusted Over time, the debt that is stored on an account's service agreement(s) may be adjusted. For more information about an adjustment, see Adjustment Details.

Bill Details

The following diagram illustrates the relationship between an account and its bills:


Accounts and their Bills
Bills and their Bill Segments
Bill Segments and their Financial Transactions
Account
Account 302291991
SA Type: Elec, Resid, Gen
SA Type: Elec, Residential
Rate: EC-1
Electric Service Agreement
SA Type: Gas, Resid, Gen
SA Type: Gas, Residential
Rate: RW-23
Gas Service Agreement
Bill
Prior Balance $100.00
Bill Corrections $0.00
Payments $100.00
Adjustments $10.00
New Charges $115.50
Current Balance $125.50
Jane Masters
18 Oak Lane
SF, CA
94514
Bill Message
We value you as a customer. Please call if you have any questions about your rate.
Electric Bill Segment
Electric Service through 1-Apr-1999
3 phase meter charge $10.00
First 300 kwh @ 0.07 $21.00
Remaining 100 kwh @ 0.09 $9.00
State tax 5% $2.00
Gas Bill Segment
Gas Service through 1-Apr-1999
First 10 therms @ $1.00 $10.00
Next 30 therms @ $2.00 $60.00
State tax 5% $3.50
BILL Electric Service Financial Transaction
A/R-residential $42.00
Electric revenue ($40.00)
State tax payable ($2.00)
Cur Amt: $42.00
Payoff Amt: $42.00
Financial Transaction
Original Financial Transaction
BILL Gas Service Financial Transaction
A/R-residential $73.50
Electric revenue ($70.00)
State tax payable ($3.50)
Cur Amt: $73.50
Payoff Amt: $73.50
BILL CANCEL Gas Service Financial Transaction
A/R-residential ($73.50)
Electric revenue $70.00
State tax payable $3.50
Cur Amt: $73.50-
Payoff Amt: $73.50-
Canceled Financial Transaction

The following concepts are illustrated above:

A bill is produced for an account Over time, many bills are produced for an account. Bills charge for the services supplied to a customer. The above illustration shows a single bill.

Bills contain bill segments A bill typically contains one bill segment for every active service agreement linked to its account. The only time this is not true is when service agreements for different frequencies exist. For example, an account with a monthly and a quarterly service agreement will only have 4 bills a year that contain both bill segments; the other months' bills will contain a single bill segment for the monthly service agreement.

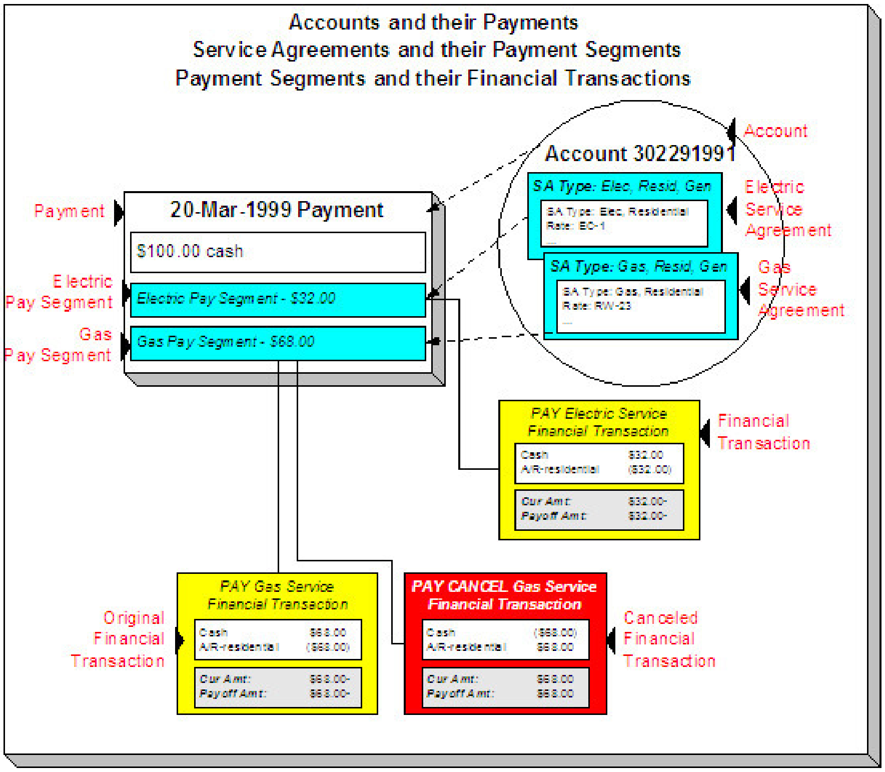
Bill segments contain calculation details A bill segment contains information showing how the segment was calculated and how it should be printed on the customer's bill.

A bill segment has a financial transaction A bill segment has a related financial transaction. A financial transaction contains the financial effects of the bill segment on the service agreement's current and payoff balances and on the general ledger.

Canceling a bill cancels the financial tran. If the bill segment is eventually cancelled, another financial transaction will be linked to the bill segment to reverse its original financial transaction. The cancellation financial transaction appears on the next bill produced for the account as a bill correction.

Payment Details

The following diagram illustrates the relationship between an account and its payments:


Accounts and their Payments
Service Agreements and their Payment Segments
Payment Segments and their Financial Transactions
Account
Account 302291991
SA Type: Elec, Resid, Gen
SA Type: Elec, Residential
Rate: EC-1
...
Electric Service Agreement
SA Type: Gas, Resid, Gen
SA Type: Gas, Residential
Rate: RW-23
...
Gas Service Agreement
Payment
20-Mar-1999 Payment
$100.00 cash
Electric Pay Segment
Electric Pay Segment - $32.00
Gas Pay Segment
Gas Pay Segment - $68.00
PAY Electric Service Financial Transaction
Cash $32.00
A/R-residential ($32.00)
Cur Amt: $32.00-
Payoff Amt: $32.00-
Financial Transaction
Original Financial Transaction
PAY Gas Service Financial Transaction
Cash $68.00
A/R-residential ($68.00)
Cur Amt: $68.00-
Payoff Amt: $68.00-
PAY CANCEL Gas Service Financial Transaction
Cash ($68.00)
A/R-residential $68.00
Cur Amt: $68.00
Payoff Amt: $68.00
Canceled Financial Transaction

The following concepts are illustrated above:

Payments are made for accounts Over time, many payments may be applied to an account's debt. The above illustration shows a single payment.

Payments contain payment segments A payment contains one payment segment for every service agreement to which the payment is distributed. For a customer who pays in full, the number of payment segments will coincide with the number of bill segments on the bill being paid.

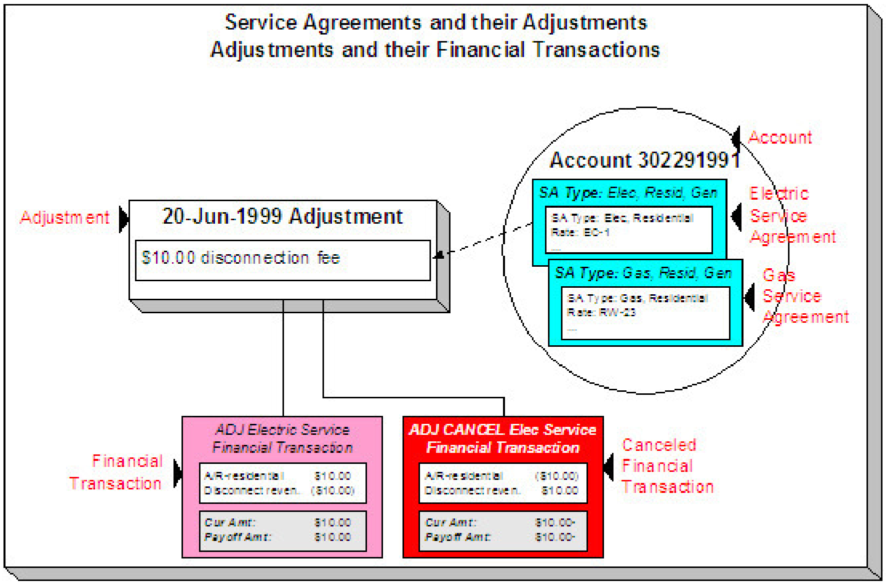
A pay. segment has a financial transaction A payment segment has a related financial transaction. A financial transaction contains the financial effects of the segment on the service agreement's current and payoff balances and on the general ledger.

Canceling a payment cancels the fin. tran. If the payment is eventually cancelled, another financial transaction will be linked to the related payment segment(s) to reverse their financial effect. The cancellation financial transaction appears on the next bill produced for the account as a negative payment.

- **FASTPATH:**
  A payment cannot be applied to an account's debt without an associated payment event. Refer to The Big Picture of Payments for more information.

Adjustment Details

The following diagram illustrates the relationship between an account and its adjustments:


Service Agreements and their Adjustments
Adjustments and their Financial Transactions
Account
Account 302291991
SA Type: Elec, Resid, Gen
SA Type: Elec, Residential
Rate: EC-1
Electric Service Agreement
SA Type: Gas, Resid, Gen
SA Type: Gas, Residential
Rate: RW-23
Gas Service Agreement
Adjustment
20-Jun-1999 Adjustment
$10.00 disconnection fee
Financial Transaction
ADJ Electric Service Financial Transaction
A/R-residential $10.00
Disconnect reven. ($10.00)
Cur Amt: $10.00
Payoff Amt: $10.00
ADJ CANCEL Elec Service Financial Transaction
A/R-residential ($10.00)
Disconnect reven. $10.00
Cur Amt: $10.00-
Payoff Amt: $10.00-
Canceled Financial Transaction

The following concepts are illustrated above:

Service agreements have adjustments Over time, a service agreement may have many adjustments. The above illustration shows a single adjustment to one of the account's service agreements.

An adjustment has a financial transaction An adjustment has a related financial transaction. The financial transaction contains the financial effects of the adjustment on the service agreement's debt and on the general ledger.

Canceling an adjust. cancels the fin. tran. If the adjustment is eventually canceled, another financial transaction will be linked to the adjustment to reverse its financial effect. The cancellation financial transaction appears on the next bill produced for the account as an adjustment.

Current Amount versus Payoff Amount

A financial transaction contains two very important attributes: payoff amount and current amount. These attributes contain the grand total of how much the customer owes.

- Current amount contains how much the customer THINKS THEY OWE.
- Payoff amount contains how much the customer REALLY OWES.

You may be wondering when these two values can be different? Well, for most financial transactions, these values are the same. These values differ under the following situations:

- When a bill segment charges a customer for a charitable contribution, payoff amount will be zero because the customer doesn't really owe anything (they don't have to contribute if they don't want to). Current amount will be equal to the agreed charitable contribution amount (the customer thinks they owe the contribution).
- When a bill segment charges a customer for a deposit, payoff amount will be zero because the customer doesn't really owe anything (billed deposits are typically not viewed as being a receivable). Current amount will be equal to the amount billed (the customer thinks they owe the deposit amount).
- When a bill segment charges a customer who participates in a levelized payment program (e.g., budget billing or non-billed budgets) the two "amounts due" will contain different values. Payoff amount is equal to how much the customer really owes for the service they consumed; current amount is equal to how much they think they owe in accordance with their monthly budget.

A perhaps easier way to view these two attributes is to consider payoff amount as the "cash out amount", i.e., the amount the customer would owe the utility if they wanted to clear up all debt. The current amount contains the amount the customer thinks they owe. If you're still struggling with the difference, think about your monthly Visa bill: it contains a monthly minimum payment and the total amount owed. The minimum payment is the current amount; the total amount owed is the payoff amount.

The topics in this section provide more information about these two fields.

What Controls What Gets Booked To Current And Payoff Amount?

As described in Bill Details, every bill segment has a sibling financial transaction. The financial transaction defines the bill segment's affect on current and payoff amounts due. The system populates these two fields as per the Financial Transaction Algorithm defined on the bill segment's bill segment type.

- **FASTPATH:**
  For more information, refer to Billing - Current Balance versus Payoff Balance and Designing and Defining Bill Segment Types.

As described in Payment Details, every payment segment has a sibling financial transaction. The financial transaction defines the payment segment's affect on current and payoff amounts due. The system populates these two fields as per the Financial Transaction Algorithm defined on the payment segment's payment segment type.

- **FASTPATH:**
  For more information, refer to Payment - Current Balance versus Payoff Balance and Setting Up Payment Segment Types.

As described in Adjustment Details, every adjustment has a sibling financial transaction. The financial transaction defines the adjustment's affect on current and payoff amounts due. The system populates these two fields as per the Financial Transaction Algorithm defined on the adjustment's adjustment type.

- **FASTPATH:**
  For more information, refer to Adjustments - Current Balance versus Payoff Balance and Setting Up Adjustment Types.

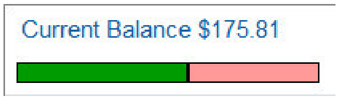
Arrears

The system keeps track of the age of each customer's debt to the day. For example, if a customer hasn't paid their last two bills, the customer's aged debt might look as follows:

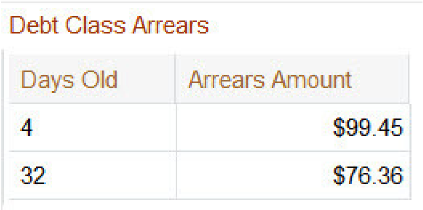
- $124.50: 22 days old
- $213.41: 51 days old

Please be aware that it is the current balance (i.e., what the customer thinks they owe) that is aged. Also keep in mind that the moment an FT is frozen, it impacts a customer's current balance.

The system represents aged debt in a variety of ways of the various transactions in the system. On the Current Context Zone and the Financial Information Zone arrears are shown in a colorful bar (where each color corresponds to different aged buckets):


Current Balance $175.81

Whereas on Service Agreement - Main, aged debt is shown in a grid:


Debt Class Arrears

Days Old	Arrears Amount
4	$99.45
32	$76.36

The grid method is used on many pages throughout the system. The following rows *may* appear in the grid:

- A row labeled New Charge highlights all debt that hasn't started aging yet. For example, if you've created a late payment charge and it hasn't appeared on one of the customer's bills, it will be classified as a New charge until the next bill is completed for the customer (unless a user overrides the late payment charge's arrears date by drilling into the financial transaction).
- A row with a label containing n Future (where n is the number of days) appears if there is "future debt". Future debt is very rare and can only exist if a debit financial transaction has a future arrears date. Financial transactions can receive a future arrears date if a bill is completed with a future date or if a user overrides a financial transaction's arrears date with a future date).
- A row that contains a number (and nothing else) represents debt that has started aging. The number is the age of the respective debt. In the above example, the customer has 1 day old debt, and debt that is more than 150 days old. Notice that the 150 day old debt is prefixed with a +. This means that the related debt is more than 150 days old. This age limit is controlled by a field on Installation Options - CC called "Oldest Bucket Age". This field limits the number of days the system will age debt. For example, if you set this field to 150, the system will never age an FT more than 150 days (and all debt that's older than 150 days will be classified as 150 day old debt). Also note, the aged debt bar that appears on Current Context Zone only ages debt a maximum of 60 days.
- A row with a label of Disputed appears if the account is an open-item customer and this customer has disputed financial transactions.

- **FASTPATH:**
  Refer to Financial Transactions And Aged Debt for more information.

GL Accounting Information

Be aware that if payoff amount is non-zero, the financial transaction has general ledger detail lines.

There are unusual financial transactions whose payoff amount is zero, but still affect the general ledger:

- Bill segments for company usage do not impact payoff amount (because your organization doesn't really owe itself anything). However, the GL is affected.
- Payment segments for charitable contributions (created when your customers contribute extra money to a charity) do not impact payoff amount. Why? Because payoff amount is never debited when a charitable contribution is billed (the

customer doesn't truly owe you for this receivable). It's only when the customer pays the contribution that the GL is impacted (debit cash, credit charitable contribution payable).

- If the SA has a special role of Loan, the financial transaction algorithms supplied with the base package transfer the current amount between the long-term receivables and the short-term receivables in the GL. This allows the general ledger to differentiate between unbilled loan receivables (long term) and billed loan receivables (short term). Refer to Payoff Balance and Current Balance for Loans for more information.

The effect on your GL is controlled by the financial transaction algorithm defined on your bill segment and payment segment types.

FASTPATH:
Refer to The GL Interface for how GL account information is interfaced to the general ledger.

A Complicated Example

The financial ramifications of a revolving charge account are predictable (if you're an accountant). The following table outlines the different financial events and their impact on the general ledger, arrearage history, and the amounts due (both current and payoff).

Event	GL Accounting	Arrearage Rule	Effect On Payoff Amt	Effect On Current Amt	Payoff Balance	Current Balance
Merchandise purchased	A/R 1000 Revenue <1000>	n/a (current amount is zero)	+1000	0	1000	0
Monthly bill	A/R 10 Int. Rev <10>	$120 aged accordingly	+10	+120	1010	120
Payment received	Cash 120 A/R <120>	$120 relieved accordingly	-120	-120	890	0

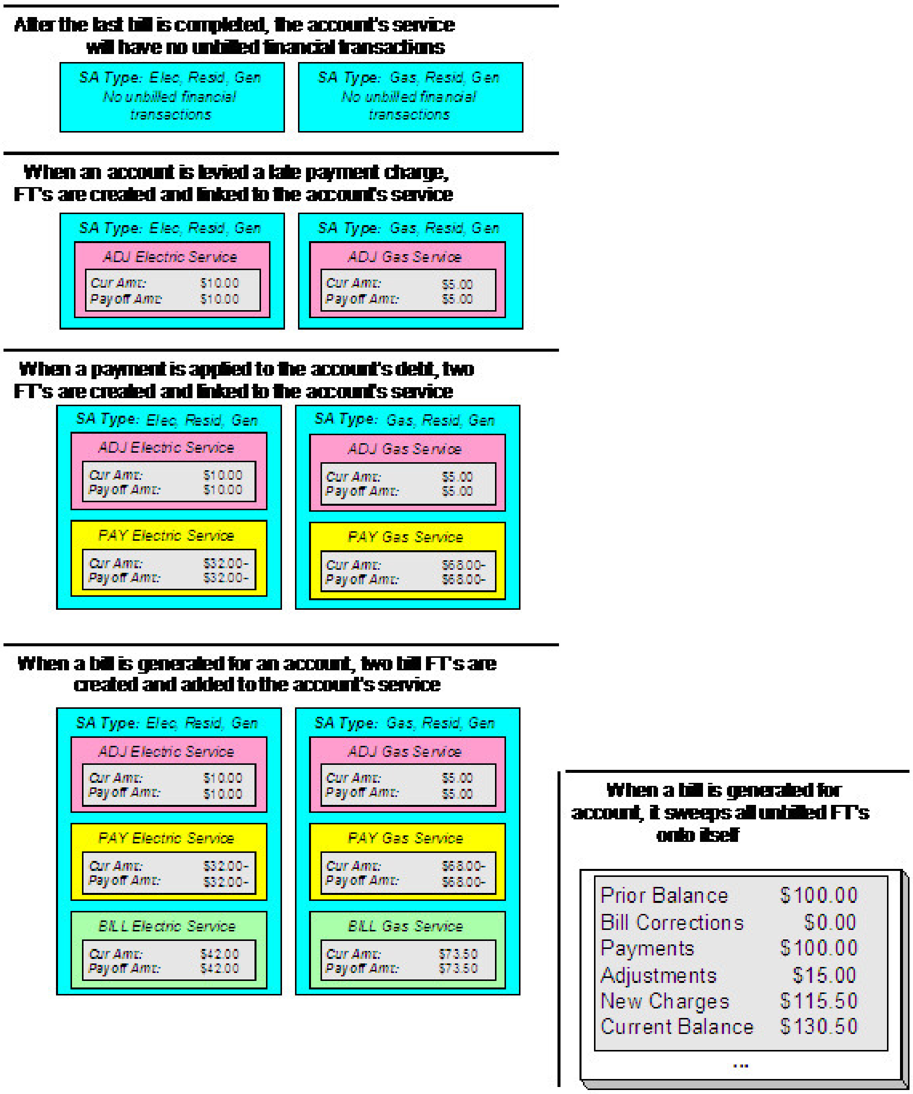
The following points describe the events in the above table:

- **Merchandise purchased.** When a customer purchases an air conditioner:
    - The system generates an adjustment to book the purchase.
    - The customer doesn't really think they owe the entire $1,000 (because they've purchased it on credit), therefore no moneys are booked to current amount. However, if the customer wanted to cash out, they would owe your organization $1,000, therefore the entire amount of the purchase is booked to payoff amount.
    - Because no money was booked to current amount, this event has no impact on the arrearage history.
- **Customer billed.** Monthly, the system calculates how much the client owes. In this example, interest is calculated to be $10 and the minimum monthly payment is set at $120.
    - The interest is posted to the GL, but principal isn't since it was booked when the merchandise was purchased.
    - The customer really thinks they owe the minimum payment amount, $120. Therefore, current amount is affected. However, if the customer were to cash out, they would owe your organization $1,000 + $10 (the interest); therefore payoff amount is affected by only $10.
    - Because current amount changed by $120, arrearage history is affected accordingly.
- **Payment received.** With any luck, the client will pay the $120 that was billed (note, they could obviously pay more).

- The payment has a normal affect on the GL (debit cash, credit A/R).
- The amount the customer thinks they owe decreases by $120, therefore current amount is affected by the payment amount. And, if the customer was to cash out, they would owe the utility $120 less, therefore payoff amount is affected by the payment amount.
- Because current amount changed by $120, arrearage history is affected accordingly.

Financial Transactions Created Between Bills

The following diagram illustrates how frozen financial transactions (FT's) accumulate between bills and are swept onto the next bill produced for the account (when the bill is completed). This example assumes



When any type of financial transaction is frozen, it impacts the related service agreement's current and payoff balances. If you do not want adjustments and bill segments to affect the customer's balance until they appear on the customer's next bill, refer to Preventing SA Balances And The GL From Being Impacted Until Bill Completion.

Notice the balances in the financial summary of the above bill:

- The **Prior Balance** is the ending balance from the customer's prior bill.
- The **Bill Corrections** portion is blank. It contains a value if you cancel / rebill a bill segment that appeared on an earlier bill.
- The **Payments** portion shows payment financial transactions (both new payments and cancellations) that have been created since the last bill.
- The **Adjustments** portion shows adjustment financial transactions (both new adjustments and cancellations) that have been created since the last bill.
- The **New Charges** portion shows bill financial transactions that were created when the bill was created.
- The **Current Balance** is the total amount owed.

●

FASTPATH:

If you practice Open Item Accounting, refer to Open Item Versus Balance Forward Accounting for more information about financial transactions and bills.

Financial Transactions And Aged Debt

The system keeps track of how old a service agreement's current balance is in order to determine if the customer is in arrears (and therefore credit and collections processing should start).

A financial transaction (FT) impacts the related service agreement's current and payoff balances the moment it is frozen. However, some types of frozen FTs have no impact on a customer's aged debt until the next bill is completed for the account associated with the service agreement.

As described in the previous section, a frozen financial transaction (FT) waits in limbo until the customer's next bill is produced. This limbo period could be several weeks if the customer is billed infrequently. When the customer's next bill is completed, all such frozen FT's are linked to the bill. It is important to stress the following in respect of these FT's:

- If the FT decreases the amount of debt, the customer's aged debt is affected immediately regardless of whether the FT appears on a bill.
- If the FT increases the amount of debt, the amount the customer owes from an aged debt perspective may or may not be affected by the FT. There is a switch on an FT called New Charge that controls the arrears behavior.
  - If this switch is on, the amount of debt will be reflected as a "new charge" when you look at the customer's aged debt. This amount will remain classified as a "new charge" until the FT is swept onto a bill. The moment the FT is swept onto the customer's bill, the debt starts aging. This logic exists because you probably don't want to start aging an FT until the customer has actually seen it.
  - If this switch is off, the date on which the FT starts aging must be defined in the Arrears Date field. The Arrears Date is used to compute how many days old the debt is.

➤

NOTE:

Aged debt limitations. It's important to be aware that there's a field on Installation Options - CC called "Oldest Bucket Age" that limits the number of days old the system will age debt. For example, if you set this field to 360, the system will never age an FT more than 360 days (and all debt that's older than 360 days will be classified as 360 day old debt). Also note, the aged debt bar that appears on Control Central - Account Information only ages debt a maximum of 60 days.

●

FASTPATH:

> If you practice Open Item Accounting, refer to Open Item Versus Balance Forward Accounting for information about how open-item FT's affect aged debt.

Preventing SA Balances And The GL From Being Impacted Until Bill Completion

It's important to understand that when any type of financial transaction is frozen, the related service agreement's balance is affected. For example:

- When a payment is frozen, the customer's balance is reduced.
- When an adjustment is frozen, the customer's balance is impacted.
- When a bill segment is frozen, the customer's balance is increased (typically).

For payments, there is no issue. However, for bill segments and certain types of adjustments, you may NOT want the customer's balance to be impacted until the next bill is completed. Consider the following scenarios:

- Late payment charges:
  - You can setup the system to create a late payment charge (i.e., an adjustment) say 5 days after an unpaid bill is due.
  - If the related adjustment is frozen, the customer's balance will be impacted. However, its impact will not affect aged debt until the next bill is completed. In other words, the amount of the frozen adjustment segment will appear as a "New Charge" until the bill is completed.
- Batch billing:
  - If a customer has multiple service agreements, it's possible for one of the service agreements to have a bill segment that's in error and the other service agreement's bill segment to be error-free.
  - If this happens and you have setup the bill cycle schedule to freeze bill segments if they're error-free, then you could have one bill segment frozen and another in error.
  - The frozen bill segment will impact the customer's balance. However, its impact will not affect aged debt until the bill is completed (and a bill cannot be completed until all of its bill segments are error-free). In other words, the amount of the frozen bill segment will appear as a "New Charge" until the bill is completed.

WARNING:
We'd like to stress that while a frozen financial transaction impacts a customer's balance the moment it is frozen, the amount of the financial transaction appears as a "New Charge" when viewing a customer's aged-debt. This amount will remain classified as a "New Charge" until the next bill is completed (i.e., the customer's debt doesn't start aging until the next bill is sent to the customer).

While aged-debt isn't impacted by frozen FT's, the general ledger is. This is because a financial transaction is marked for interface to the general ledger when it is frozen. This can be problematic if you have a long period between FT freeze and bill completion (you could impact the general ledger but not impact the customer's balance). If this is unacceptable, you can setup the system to not allow certain types of FT's to be frozen until the next bill is completed. This means that neither the customer's balance nor the general ledger will be impacted until bill completion time. To do this:

- Choose the Freeze At Bill Completion option on Installation Options - Billing.
- Examine each of your adjustment types. Select Freeze At Bill Completion for those that should not impact the customer's balance or the general ledger until the next bill is completed. Select Freeze At Will for those that should impact the customer's balance and the GL when they are frozen. Typically, the only adjustment types for which you'd choose Freeze At Will option are those that cause a customer's balance to be reduced, those that are used to refund money to a customer, and those that are created at bill completion. Adjustment types for adjustments created during bill completion

(e.g., by a bill completion algorithm) must have their adjustment freeze option set to Freeze At Will. Otherwise (i.e., if the option is Freeze At Bill Completion) they will not be frozen until a subsequent bill is completed.

Please be aware of the following in respect of the Freeze At Bill Completion options:

- If you turn on Freeze At Bill Completion on Installation Options - Billing:
  - Users will not be allowed to freeze bill segments online. This means that the freeze button will be disabled on Bill - Main, Bill - Bill Segments and Bill Segment - Main.
  - The Billing background process will not freeze bill segments until all segments on a bill are error free (and permission has been granted on the bill cycle schedule to complete bills).
  - Bill segments will exist in the freezable state until the bill is completed.
- If you turn on Freeze At Bill Completion on Adjustment Type - Main:
  - Users will not be allowed to freeze adjustments of this type online. This means that the freeze button will be disabled on Adjustment - Main.
  - Background processes that create adjustments will not freeze this type of adjustment. Rather, the adjustments will be frozen when the next bill is completed.
  - Adjustments of this type will therefore exist in the freezable state until the next bill is completed.

> ➤ **NOTE:**
> **Alerts highlight freezable FT's.** Please be aware that messages appear in the Account Information - Financial Information Zone and in the Dashboard - Financial Information Zone to highlight the existence of freezable financial transactions.

Please be aware of the following in respect of the Freeze At Will options:

- If you turn on Freeze At Will on Installation Options - Billing:
  - Users will be allowed to freeze bill segments online. This means that the freeze button will be enabled on Bill - Main, Bill - Bill Segments and Bill Segment - Main.
  - The Billing background process will freeze bill segments when the individual segment is error-free (and permission has been granted on the bill cycle schedule to freeze bill segments).
  - Bill segments will exist in the frozen state regardless of whether the bill is completed.
  - The frozen bill segment's FT will be interfaced to the GL when the interface next runs.
  - All adjustment types must be also be set to Freeze At Will (otherwise they wouldn't get frozen).
- If you turn on Freeze At Will on Adjustment Type - Main:
  - Users will be allowed to freeze adjustments of this type online. This means that the freeze button will be enabled on Adjustment - Main.
  - Background processes that create adjustments will freeze this type of adjustment.
  - Adjustments of this type will exist in the frozen state prior to bill completion.
  - The frozen adjustment's FT will be interfaced to the GL when the interface next runs.

Forcing The Freeze Date To Be Used As The Accounting Date

Every financial transaction references an accounting date. The accounting date controls the accounting period to which the financial transaction is booked as described below:

- Every financial transaction references an accounting date and a service agreement
- Every service agreement references a service agreement type
- Every service agreement type references a GL division
- Every GL division references an accounting calendar
- The accounting calendar contains the cross-reference between the accounting date specified on the financial transaction and the related accounting period in your general ledger

The accounting date is populated on financial transactions when they are initially generated. The following points describe the source of the accounting date:

- The user who creates or cancels a bill segment online defines the accounting date as part of the generation/cancel dialog (note, the current date defaults).
- Bill segments that are produced by the BILLING background process have their accounting date defined on the bill cycle schedule that caused the bill to be created.
- The user who creates or cancels an adjustment online defines the accounting date as part of the generation / cancel dialog (note, the current date defaults).
- Payments are unusual in that their financial transaction is only created when they are frozen (rather than when the payment is first distributed amongst the account's service agreements). At payment freeze time, the accounting date is set to the current date.

For payments, there is no issue because the accounting date is only populated on the financial transaction when a payment is frozen. However, for bill segments and adjustments, your business practice may dictate that the freeze date should be used as the accounting date rather than the original accounting date. Alternatively, your business practice may dictate that the accounting date that's originally stamped on bill segments / adjustments should be used (unless this associated period is closed at freeze time). It's really a question of the interpretation of the local accounting rules. After you've decided on your approach, populate the **Accounting Date Freeze Option** on Installation Options - Billing with one of the following values:

- Choose Always change if the accounting date on your financial transactions should be populated with the freeze date (i.e., the current date when the financial transaction is frozen).
- Choose Change if period is closed if the accounting date defined when the financial transaction is generated should be used (unless the associated accounting period is closed).

Please be aware of the following in respect of your choice:

- If you choose Always change:
  - When a user freezes a bill segment online, they will be prompted to supply an accounting date. The current date will default, but the user can override this value.
  - When a user freezes an adjustment online, they will be prompted to supply an accounting date. The current date will default, but the user can override this value.
  - The BILLING background process will use the current business date as the accounting date on bill segments that it freezes.
  - Also note, if you have chosen the Freeze At Bill Completion **Bill Segment Freeze Option** on the installation record, bill segments and certain types of adjustments are frozen when a bill is completed. This means that the accounting date on the related financial transactions will be set to the completion date (because the completion date is the freeze date with this setting). Refer to Preventing SA Balances And The GL From Being Impacted Until Completion for more information.
- If you choose Change if period is closed:
  - When a user freezes a bill segment online, they will only be prompted to supply an accounting date if the related accounting period is closed (because the accounting period closes after the bill segment is generated but before it's frozen). The current date will default, but the user can override this date.

- When a user freezes an adjustment online, they will only be prompted to supply an accounting date if the related accounting period is closed (because the accounting period closes after the adjustment is generated but before it's frozen). The current date will default, but the user can override this date.
- The BILLING background process will use the accounting date defined on the related bill cycle schedule as the accounting date on the bill segments that it creates and freezes.

NOTE:
The above installation option only controls the final accounting date for GL recording purposes. Rate and bill factor value selection based on accounting date uses the date as initially determined.

How Late Payment Charges Get Calculated

Late payment charges are system-generated adjustments used to penalize a customer for late (or no) payments. This section describes how to set up the tables that control how and when late payment charges are generated. The following points describe how and when late payment charges are calculated.

- When a bill is completed, the system marks it with the date on which late payment charges will be calculated if the bill is not paid.
  - This date is calculated by adding grace days to the bill's due date. Grace days are defined on the account's Customer Class / Division.
  - This date will be zero if the account's Customer Class / Division indicates the account is not eligible for late payment charge processing.
- The late payment charge background process (referred to by the batch ID of **LATEPYMT**) selects all bills on or after their late payment charge date.
  - For each such bill, the system determines if its account satisfies the late payment charge eligibility criteria defined on the account's Customer Class / Division. The eligibility criteria are defined in an algorithm and can therefore be as flexible as required.
  - If an account is eligible for late payment charges, the system checks each of the account's service agreements to determine if it is eligible for late payment charges (as defined on the service agreement's SA Type).
  - If a service agreement is eligible for late payment charges, the system calls the SA type's late payment charge calculation algorithm. This algorithm should calculate the late payment charge amount, if applicable and return the calculated amount and an appropriate adjustment type to use. If this algorithm returns this information, an adjustment is generated to levy the late payment charge.

FASTPATH:
Refer to Setting Up Customer Classes for more information about how to set up an account's due days and grace period. Refer to SA Type - Main Information for more information about enabling late payment charges calculations for your service agreements.

You can update the **Late Payment Charge Details** section on the Bill - Main Information page to indicate if and when late payment charges may be levied. For more information, see Bill - Main Information.

Service Agreement Type Controls Everything

The previous section illustrated three important concepts:

The true financial impact of the three financial events - bills, payments, adjustments - is at the service agreement level, not at the account level. This means that bills and payments are meaningless on their own. It's the service agreements' bill segments, payment segments and adjustments that affect how much a customer owes.

- Every bill segment, payment segment, and adjustment has a related financial transaction. These financial transactions contain the double-sided journal entries that will be interfaced to your general ledger. They also contain the information defining how the customer's debt is affected by the financial event (i.e., current amount and payoff amount).
- A single bill can contain many bill segments, each of which may have a different frequency. For example, a bill could contain future charges, monthly retroactive charges based on measurement cycle, quarterly charges that must end on a quarter-end boundary.

You control the financial effects of the various financial events using a single field on the service agreement. This field is called the service agreement (SA) Type. In this section, we describe many of the tables that must be set up before you can create a SA type.

➤

NOTE:
Foreshadowing. You will notice that we don't explain how to set up SA types in this section. This is because SA type controls numerous aspects of a service agreement's behavior in addition to its financial behavior. The non-financial aspects are discussed in later chapters. It's only after you have set up all of the control tables in this manual that you'll be able to finally define your SA types. Refer to Setting Up SA Types for more information.

WARNING:
Take the time to define how you will record the various financial events in your general ledger before you attempt to set up these control tables. If you have simple accounting needs, this setup process will be straightforward. However, if you sell many services and use sophisticated accounting, this setup process will require careful analysis.

Understanding CIS Divisions

There are two types of Divisions referenced on a SA type: a CIS Division and a GL Division. This is a rather powerful structure, but it can be confusing.

- General Ledger divisions typically comprise individual entities (e.g., companies) in your general ledger. You must set up a GL division for each such entity. The GL division's sole purpose in the system is to define the accounting period associated with financial transactions linked to service agreements associated with the GL division (service agreements are associated with GL divisions via their SA type). The system cares about accounting periods in order to prevent a user from booking moneys to closed periods. It also uses accounting periods when it produces the flat file that contains the consolidated journal entry that is interfaced to your general ledger (refer to The GL Interface for more information).
- A CIS division is associated with a jurisdiction. The definition of a jurisdiction is a geographic-oriented entity with unique business rules. For example, if you conduct business in California and Nevada, and each state has different collection rules, you will need a separate jurisdiction for each state. You must set up a CIS division for each jurisdiction in which you conduct business.

- CIS division is also referenced on service agreement, premise and account.
  - The CIS division on SA is actually part of the SA's SA type. Because SA type controls many business rules, all business rules that are on the SA type can be thought of as being defined for a given jurisdiction and SA type combination. For example, you could define your valid rates for electric residential service in California which differ from the valid rates for electric residential service in Nevada. Refer to Defining Service Agreement Types for more information. In addition to controlling the business rules defined on the SA's SA type, the SA's CIS division also controls the type of collection criteria used to determine if and how to collect overdue debt. Refer to Setting Up Collection Class Control for more information.
  - The CIS division on premise defines the jurisdiction in which the premise is located. This jurisdiction controls the types of service agreements that can be associated with the premise's service points (e.g., you can only link California-oriented service agreements to premises governed by the California jurisdiction). You can also set up your field activity types to execute special algorithms when a field activity is completed at a service point located in specific jurisdiction.
  - The CIS division on account when combined with the account's customer class defines the jurisdiction that governs financial business rules (e.g., the bill's due date, when and how late payment charges are calculated, etc.). Refer to Setting Up Customer Classes for more information about these rules. The CIS division on account can also play a part in the addressee of To Do entries associated with the account. To assign To Do entries to a role based on the division, simply link the To Do type to the division. Refer to To Do Entries Reference A Role for more information.

CIS Division Overrides

Some configuration values in the system, such as those defined on the installation record, are global and applied to the entire system. CIS Division allows some of these values to be specified by CIS division, overriding the system-wide default configuration. The system processes check the CIS Division for the override before checking the default configuration. CIS Division allows for overrides to be specified for **customer class** (by person and business), and **credit rating** and **cash-only** beginning and threshold values.

The bill route type and quote route type are used to define the background process and extract algorithm for each route type. CIS Division allows overrides to be specified for the extract batch control and/or extract algorithms. These can be defined for any or all each bill route types and quote route types. While bill route type and quote route type allow many of each type to be configured, one of each type is defaulted from the installation record. This override allows CIS Division specific value for a common route type.

Users and CIS Division

When the installation option Control By CIS Division flag is set, it is necessary to associate the system users with CIS Division. This feature is applicable if your organization expects users to work on designated CIS Divisions only (as opposed to allowing users to serve customers across all CIS Divisions). Users can be associated with CIS Division(s) even if the above flag is not set. Some system functions take advantage of this such as defaulting CIS Division on control central searches. User - CIS Division is maintained in the user portal.

To navigate there, select **Admin** > **User** > **Search** and navigate to the **Miscellaneous** page.

How Does the System Use User CIS Division

See Installation Options – Main for information on how this is used in conjunction with the Control By CIS Division flag

The user’s CIS Divisions are used in some circumstances to restrict case type and customer contact type by CIS Division.

The user’s Primary CIS Division is used to default CIS Division on the control central search is provided with the system.

Restricting CIS Division

CIS Division appears through much of the system. The primary objects that reference a CIS Division are account, service agreement, and premise. It is also referenced on secondary objects such as orders and on pages that interact with the primary objects such as start/stop and payment arrangement.

A flag on CIS Division, Restrict to Account CIS Division, can be used to restrict service agreements and premise to the account's CIS Division. It is important to note that it is the account's CIS Division that matters. When this flag is set, many of the system pages default CIS Division and restrict the user for changing the value. The following rules are enforced in addition to the visible changes:

- Service agreement's CIS Division must match its account's CIS Division.
- Account CIS Division cannot be changed if the account has non-canceled service agreements.
- Account Protect CIS Division switch is turned on and is protected. This is done because the switch becomes irrelevant since all service agreements match the account's CIS Division and the account CIS Division would not change.
- When starting service, only premises that match the account CIS Division or do not have a CIS Division maybe chosen. This rule is applied when choosing a premise via start/stop or order.
- While most SA/SPs are created by the system when service is started, it is possible to manually maintain SA/SPs. When searching for a service point from Service Agreement - SA/SP, the search is limited to SPs whose premise matches the account CIS Division or does not have a CIS Division.
- As mentioned, this setting applies to the account CIS Division. When changing a premise's CIS Division, the system retrieves accounts that have been associated with the premise via the premise's SPs to SA/SPs to service agreement to account. For any account found, that account's CIS Division is checked to see if the restrict CIS Division flag has been set. If found, the premise CIS Division is restricted to that CIS Division or blank of the premise type permits.

  ➤ **NOTE:** It is possible for a premise to have been associated with many accounts over time and those accounts can have different CIS Divisions if the CIS Division does not have the restrict CIS Division flag set. The system should prevent the premise from being associated with more than one account with different CIS Divisions with the restrict CIS Division flag set as long as the restrict CIS Division flag is set before the data is created. As noted above, setting the flag with existing data can result in undesired issues.

- The characteristic premise on a service agreement is restricted to premises linked to the account if the service agreement requires an SP. For other service agreements the premise search is restricted to the account CIS Division.

WARNING: Do not turn on this flag without reviewing your existing data to ensure that it adheres to the rules enforced when the flag is set. The above rules are enforced on not just current data, but historic data as well such as closed service agreements. It is intended to ensure a consistency over the life of the objects associated with the CIS Division. This flag is configurable for each CIS Division, but consider the implications of setting it on some, but not all CIS Divisions.

Using CIS Division for Operational Segmentation

CIS Division is found throughout many of the entities in the system and there are many configuration settings related to CIS Division. When these configuration settings are used in conjunction with each other, it is possible to operate distinct business entities or jurisdictions that are segmented by CIS division. The following summary describes the different settings and uses of CIS Division that you can configure if your organization operates in multiple jurisdictions and needs to segment data and business processes by CIS Division. Details about each of these can be found in the section of the document where the functionality is configured.

Associate CIS Divisions with Users

CIS Divisions can be associated with users and one can be denoted as the primary.

Control By CIS Division

Setting the Control By CIS Division flag on the installation record drives many system changes. On the account it makes CIS Division required. The flag drives functionality changes on other pages such as person, premise, and order.

Premise/SP Access

Setting the Premise/SP Restrict flag on the installation record restricts access to premise and SPs not link to an account. The access is restricted to the user’s CIS Division(s).

CIS Division

The CIS Division entity itself, allows for the configuration of a work calendar, default customer classes for persons and businesses, credit rating and cash only beginning and threshold values, To Do role assignment, and bill and quote route extract batch and algorithm overrides.

Restrict CIS Division

The Restrict to Account CIS Division flag is a setting on CIS Division. It is used to ensure that service agreements and premises match the account CIS Division and is a main component of segmenting data by CIS Division.

Customer Class

Customer class is used to default collection class on accounts.

Customer Class Control

Customer Class Control defines many business rules directly or through algorithms. These rules are specified per CIS Division. As noted above, Customer Class can be defaulted from CIS Division so there is a lot of flexibility in the configuration of these two entities. Some of the business rules are defaults such as access group and budget plan. Others such as payment distribution and billing days until due define the rule that apply to the account Customer Class/CIS Division combination.

SA Type

CIS Division is part of the SA Type key. SA Type defines the valid values for SAs of that type and also defines attributes such as debt class and write off debt class for of SAs of that type.

Collections

As noted above, Collection Class is defaulted on the account from customer class and SAs inherit their debt class from their SA Type. Collection Class Controls are defined for the unique combination of collection class and debt class. Collection class controls define the business rule and collection processes. Through configuration, collection rules can be defined for specific CIS Divisions.

Write Off

As noted above, collection class is defaulted on the account from customer class and SAs inherit their write off debt class from their SA Type. Write off controls are defined for the unique combination of collection class and write off debt class. Write off control define the business rule and write off processes. Through configuration, write off rules can be defined for specific CIS Divisions.

Premise Type

Premise Types can control if CIS Divisions are required, optional or not allowed for premise of that type. When starting service, if the premise has a premise type, only SA Types with the same CIS Division as the premise that are identified to start for SP Types for SPs at the premise will start. This allows more generic SP Types that start SA Types specific for the premise CIS Division.

Restricting Types by CIS Division

Adjustment type, budget plan, bill cycle, case type, and customer contact type allow a CIS Division to be specified if the type is used for a specific CIS Division. Types without a CIS Division are not restricted.

Background Processing By CIS Division

Batch billing and the collection, severance, and write off background processes can all be configured to select records by CIS Division. See Choosing Which Records Are Selected for more information

Setting Up CIS Divisions

This portal is used to display and maintain CIS Divisions.

Refer to Understanding CIS Divisions for more information.

You can access the portal from **Admin** > **General** > **CIS Division**.

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

- **CIS Division List:** This zone lists all CIS division records. Broadcast a record to display the details of the selected record.
- **CIS Division:** This zone provides information about the selected CIS Division.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference CI_CIS_DIVISION.

Setting Up Revenue Classes

Every service agreement references a service agreement (SA) type. Amongst other things, the SA type defines a service agreement's revenue class. The revenue class is used when the service agreement's rate books revenue to different GL distribution codes based on the service agreement's revenue class.

> **FASTPATH:**
> See Designing Calculation Groups and Rules for more information about how revenue class is used to determine the GL revenue accounts referenced on a bill. See Revenue Segmentation for more information about how revenue class affects the number of SA types you will need.

To set up revenue classes, choose **Admin** > **Financial** > **Revenue Class**.

Description of Page

Enter an easily recognizable **Revenue ClassID** and **Description** for every revenue class.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference CI_REV_CL.

Setting Up Distribution Codes

Distribution codes simplify the process of generating accounting entries by defining valid combinations of chart of account field values.

> **FASTPATH:**
> Refer to The Source Of GL Accounts On Financial Transactions for more information about the accounting entries associated will bills, payments and adjustments.

To set up distribution codes, open **Admin** > **Financial** > **Distribution Code** > **Add.**

Description of Page

Enter a unique **Distribution Code** and **Description** for the distribution code.

If this distribution code is a holding account used for payables cash accounting, check the **Use For Non-Accrual Accounting** switch and select the accounting method from the **Accounting Method** list. Select the priority level for the distribution code from the **Accounting Priority** list and enter the actual payable **Accounting Code.** The system will transfer monies from the holding account to the distribution code when the cash event occurs. Transfers will occur based on priority and debt age. For more information, refer to Payables Cash Accounting and Deferred Accrual Accounting.

Define the **GL Account Algorithm** used by the system when it interfaces financial transactions that reference this distribution code to your general ledger (refer to GLDL - Create General Ledger Download for more information about the download process). The logic embedded in this algorithm constructs the actual GL account number. If you haven't done so already, you must set up this algorithm in the system. To do this:

- Create a new algorithm (refer to Setting Up Algorithms).
- On this algorithm, reference an Algorithm Type that constructs your general ledger account number. Click here to see the algorithm types available for this plug-in spot.

The **Write Off Controls** control how the system writes off debt associated with the distribution code. Refer to The Ramifications of Write Offs in the General Ledger for an explanation of how these fields are used at write-off time.

- Define the **Division** and **SA Type** of the service agreement to which bad debt associated with this distribution code should be transferred at write-off time. Note: only SA Types with a special role of Write Off may be selected.
- When the system transfers debt to the write-off service agreement defined above, the distribution code defined on this **Division / SA Type** will be debited unless you turn on the **Override Switch**. When this switch is turned on, the system overrides the distribution code of the transfer to side of the adjustment with the distribution code associated with the debt being written off. You'd typically turn this switch on for liability distribution codes because you want to debit the original liability account when the debt is written off. Note: if this switch is on the system also overrides the characteristic type / value with the respective value associated with the debt that is being written off.

Use the **GL Account Details** scroll to define how the system constructs the GL account associated with the distribution code when it interfaces the financial transaction to your general ledger. For each distribution code, enter the following information:

- Enter the **Effective Date** of the following information.
- Define whether, on the **Effective Date**, the following information is Active or Inactive. The system will only use effective-dated information that is Active.
- Enter the **GL Account** that the general ledger uses to process financial transactions tagged with this distribution code.

- Enter the **Statistics Code** that should be passed to the general ledger during the GL interface for this **GL Account**. For example, if this **Distribution Code** is used to record electric, residential revenue, the **Statistics Code** would be kWh if you record the number of kWh in you general ledger along with the dollar value of the revenue.
- If you have configured your installation options to indicate that fund accounting is practiced, define the **Fund** associated with this distribution code. If your installation options indicate that fund accounting is not practiced, the field is not visible.
- Use the characteristics grid to define characteristic values for the **Distribution Code**. This grid supports all standard characteristic types (such as, Adhoc, Predefined, File Location, and Foreign Key) from the **Distribution Code GL Account Characteristics** table.. To modify a characteristic, simply move to a field and change its value. The following fields display:
    - **Characteristic Type**. Indicate the type of characteristic.
    - **Characteristic Value**. Indicate the value of the characteristic.

> **NOTE:**
> You can only choose characteristic types defined as permissible on the distribution code record. Refer to Setting Up Characteristic Types & Their Values for more information.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference CI_GL_DIVISION.

Setting Up Billable Charge Templates

A user creates a billable charge whenever a customer should be levied an ad hoc charge. For example, you would create a billable charge to charge a contractor for the repair of a ruptured gas line.

> **NOTE:**
> **Interfacing billable charges from an external system.** In addition to being entered manually, billable charges can also be interfaced from an external system. You would interface billable charges if your organization provides "pass through" billing services for a service provider. Refer to Uploading Billable Charges for more information.

A billable charge must reference a service agreement. This service agreement behaves just like any other service agreement:

- **Bill segments are created for the service agreement.** Whenever billing is performed for an account with billable charge service agreements, the system creates a bill segment for each unbilled billable charge.
- **Payments are distributed to the service agreement.** Payments made by an account are distributed to its billable charge service agreements just like any other service agreement.
- **Overdue debt is monitored.** The credit and collections process monitors billable charge service agreements for overdue debt and responds accordingly when overdue debt is detected.

> **NOTE:**
> **Rates can be applied to billable charges**. Billable charges can be connected to a service agreement that also specifies a rate. The rate will be applied and lines added to the bill segment after the billable charge lines

are added. For example, a rate can insert flat charges or be applied to service quantities associated with the billable charge.

Taxes on top of billable charges. Rates cannot be applied to billable charge lines. If you need to perform a calculation such as applying taxes on top of the existing lines, add a service quantity (SQ) that contains the taxable amount with an SQ identifier that describes it as a taxable amount. A calculation rule can apply the tax to this SQ.

●

FASTPATH:
Refer to How To Create A One-Time Invoice for instructions describing how to create a bill for a billable charge outside of the normal bill creation process.

Billable charge templates exist to minimize the effort required to create a billable charge for a customer. A billable charge template contains the default bill lines, amounts and distribution codes used to levy a one-off charge.

The information on the template may be overridden by a user when the billable charge is created. For example, you can create a billable charge template to levy tree-trimming charges. This template would contain the bill lines, amounts and distribution codes associated with a tree trimming activities. Then, when you trim a tree for a customer, a user can create a billable charge using the template and override the amount to reflect the actual amount (if it differs from the norm).

➤

NOTE:
Templates aren't required. A billable charge can be created without a template for a truly unexpected charge.

After setting up the billable charge templates, you must indicate the SA types that can use each template. Obviously, only billable charge SA types (as defined on the SA type's special role) will reference billable charge templates.

Billable Charge Template - Main

Open **Admin** > **Billing** > **Billable Charge Template** > **Add** to define your billable charge templates.

Description of Page

Enter a unique **Billable Charge Template** ID and **Description** for the billable charge template.

Use **Description on Bill** to define the verbiage that should print on the customer's bill above the billable charge's line item details.

Use **Currency Code** to define the currency in which the billable charge's amounts are expressed.

Use the grid to define the line item details associated with the billable charge (note, the **Total Line Amount** field is automatically calculated. It is the sum of the **Charge Amount** on each of the Line Sequence items). The following fields are required for each entry in the grid.

Sequence Line sequence controls the order in which the line items appear on the bill segment.

Description on Bill Specify the verbiage to print on the bill for the line item.

Charge Amount Specify the default amount to charge for the line item.

Show on Bill Turn this switch on if the line item should appear on the customer's printed bill. It would be very unusual for this switch to be off.

Appears in Summary Turn this switch on when the amount associated with this line also appears in a summary line.

Memo Only, No GL Turn this switch on when the amount associated with this line does not affect the GL (or the total amount owed by the customer).

Distribution Code Specify the default distribution code associated with this line item.

If you use the drill down button on the left most column in the grid, you will be taken to the Line Characteristics tab with the selected line displayed.

- **FASTPATH:**
  For more information about creating a billable charge, refer to Maintaining Billable Charges. For more information about billing billable charges, refer to How To Create A One-Time Invoice.

Billable Charge Template - Line Characteristics

Open **Admin** > **Billing** > **Billable Charge Template** > **Search** and navigate to the **Line Characteristics** page to define your billable charge templates line characteristics.

Description of Page

The **Line Sequence** scroll defines the billable charge template line to which you wish to assign characteristic values.

The characteristics grid supports all standard characteristic types (such as, Adhoc, Predefined, File Location, and Foreign Key) from the Billable Charge Template Line Characteristics table.

To modify billable charge template line characteristics, simply move to a field and change its value. To add characteristics, press + to insert a row and then fill in the information for each field. The following fields display:

Characteristic Type The type of characteristic.

Characteristic Value The value of the characteristic.

Billable Charge Template - SQ Details

Open **Admin** > **Billing** > **Billable Charge Template** > **Search** and navigate to the **SQ Details** page to define your billable charge templates service quantities.

Description of Page

To modify a template's service quantity, simply move to a field and change its value. To add a new service quantity to the billable charge template, press the + button to insert a row and fill in the information for each field. The following fields display:

Sequence Specify the sequence number of the SQ.

UOM Select the unit of measure of this SQ. One or more of UOM, TOU, or SQ identifier must be selected.

TOU Select the time of use period.

SQ Identifier Select the SQ identifier.

Service Quantity Specify the number of units of this service quantity.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference CI_B_CHG_TMPLT.

Designing and Defining Bill Segment Types

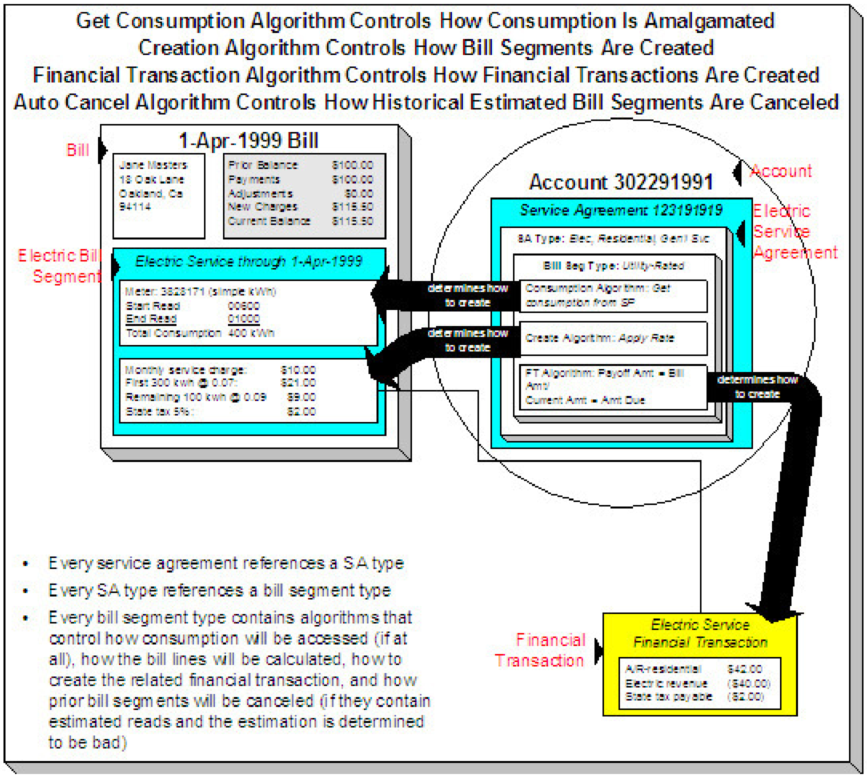
Every service agreement references a service agreement (SA) type. Amongst other things, the SA type references a bill segment type. The bill segment type controls how bill segments and their related financial transactions are created.

WARNING:
We strongly recommend understanding the concepts described in The Big Picture of Billing before setting up your bill segment types.

The topics in this section describe how to design and set up bill segment types.

What Do Bill Segment Types Do?

Bill segment types control how bill segments and their related financial transactions are created. The following illustration will help you understand how the system uses bill segment types during the bill segment creation process:



Get Consumption Algorithm Controls How Consumption Is Amalgamated
Creation Algorithm Controls How Bill Segments Are Created
Financial Transaction Algorithm Controls How Financial Transactions Are Created
Auto Cancel Algorithm Controls How Historical Estimated Bill Segments Are Canceled

- Every service agreement references a SA type
- Every SA type references a bill segment type
- Every bill segment type contains algorithms that control how consumption will be accessed (if at all), how the bill lines will be calculated, how to create the related financial transaction, and how prior bill segments will be canceled (if they contain estimated reads and the estimation is determined to be bad)

Bill
1-Apr-1999 Bill
Jane Masters
18 Oak Lane
Oakland, Ca
94114
Prior Balance $100.00
Payments $100.00
Adjustments $0.00
New Charges $115.50
Current Balance $115.50
Electric Bill Segment
Electric Service through 1-Apr-1999
Meter: 3828171 (simple kWh)
Start Read 00600
End Read 01000
Total Consumption 400 kWh
Monthly service charge: $10.00
First 300 kwh @ 0.07: $21.00
Remaining 100 kwh @ 0.09 $9.00
State tax 5%: $2.00
Account 302291991
Account
Service Agreement 123191919
Electric Service Agreement
SA Type: Elec, Residential, Gen'l Svc
Bill Seg Type: Utility-Rated
Consumption Algorithm: Get consumption from SP
Create Algorithm: Apply Rate
FT Algorithm: Payoff Amt = Bill Amt/ Current Amt = Amt Due
determines how to create
determines how to create
determines how to create
Financial Transaction
Electric Service Financial Transaction
A/R-residential $42.00
Electric revenue ($40.00)
State tax payable ($2.00)

Designing Your Bill Segment Types

The following table contains a subset of the SA types listed under Defining Service Agreement Types and Designing Your SA Types And Start Options For Sub SAs and Designing SA Types For Service Provider Financial Settlements. However, if you are reading this document from top to bottom, you probably don't know what your SA types are (they are only designed much later) and will have to forestall this task until that time.

We're going to cheat and assume you know what your SA types are and fill in the algorithms necessary to create bill segments for each SA type. After this table is complete, we will look for unique combinations of the 4 algorithms and create a bill segment type for each one.

NOTE:
Before you can fill in the columns for your own SA types, you should be comfortable with the descriptions of the algorithms described under Setting Up Bill Segment Types.

Div. SA Type	Calculation Algorithm	FT Algorithm	Consumption Algorithm	Auto Cancel Algorithm
CA/G-RES	Apply Rate To Usage Request	Payoff = Bill Amount / Current = Amount Due	Get Consumption Using Usage Request	NA - can't estimate consumption.
CA/G-COM	Apply Rate To Usage Request	Payoff = Bill Amount / Current = Amount Due	Get Consumption Using Usage Request	NA - can't estimate consumption.
CA/G-IND	Apply Rate To Usage Request	Payoff = Bill Amount / Current = Amount Due	Get Consumption Using Usage Request	NA - can't estimate consumption.
CA/CABLE	Apply Rate To Usage Request	Payoff = Bill Amount / Current = Amount Due	Get Consumption Using Usage Request	NA - can't estimate consumption.
CA/E-COY	Apply Rate To Usage Request	Payoff = 0 / Current = 0	Get Consumption Using Usage Request	NA - can't estimate consumption.
CA/E-RESU	Apply Rate To Usage Request	Payoff = Bill Amount / Current = Amount Due	Get Consumption Using Usage Request	NA - can't estimate consumption.
CA/E-COMU	Apply Rate To Usage Request	Payoff = Bill Amount / Current = Amount Due	Get Consumption Using Usage Request	NA - can't estimate consumption.
CA/WO-STD	N/A - non billable	N/A - non billable	N/A - non billable	N/A - non billable
CA/WO-LIA	N/A - non billable	N/A - non billable	N/A - non billable	N/A - non billable
CA/CHARITY	Recurring Charge	Payoff = 0 / Current = Bill Amount	N/A - no consumption is needed	N/A - no consumption is needed
CA/PA-REGU	Recurring Charge With Auto Stop	Payoff = 0 / Current = Bill Amount	N/A - no consumption is needed	N/A - no consumption is needed
CA/MERCH-I	Recurring Charge For Amount To Bill	Payoff = 0 / Current = Bill Amount	N/A - no consumption is needed	N/A - no consumption is needed
CA/DEP-I	Recurring Charge For Amount To Bill	Payoff = 0 / Current = Bill Amount	N/A - no consumption is needed	N/A - no consumption is needed
CA/ONETIME	Billable Charge	Payoff = Bill Amount / Current = Amount Due	N/A - no consumption is needed	N/A - no consumption is needed
CA/OVR UNDR	N/A - non billable	N/A - non billable	N/A - non billable	N/A - non billable
CA/E-SUB ENR	Apply Rate	Payoff = Bill Amount / Current = Amount Due		
CA/E-SUB BC	Billable Charge	Payoff = Bill Amount / Current = Amount Due	N/A - no consumption is needed	N/A - no consumption is needed
CA/E-FIN SETTL	N/A - non billable	N/A - non billable	N/A - non billable	N/A - non billable

Now, we'll extract unique combinations of the four algorithms and create a bill segment type for each.

Bill Segment Type	Calculation Algorithm	FT Algorithm	Consumption Algorithm	Auto Cancel Algorithm
BD RATED	Apply Rate To Usage Request	Payoff = Bill Amount / Current = Amount Due	Get Consumption Using Usage Request	NA - can't estimate consumption.
BILLCHRG	Billable Charge	Payoff = Bill Amount / Current = Amount Due	N/A - no consumption is needed	N/A - no consumption is needed
COMPUSAG	Apply Rate To Usage Request	Payoff = 0 / Current = Bill Amount	Get Consumption Using Usage Request	N/A - no consumption is needed
RECUR	Recurring Charge	Payoff = 0 / Current = Bill Amount	N/A - no consumption is needed	N/A - no consumption is needed
RECUR AS	Recurring Charge With Auto Stop	Payoff = 0 / Current = Bill Amount	N/A - no consumption is needed	N/A - no consumption is needed
RECURATB	Recurring Charge For Amount To Bill	Payoff = 0 / Current = Bill Amount	N/A - no consumption is needed	N/A - no consumption is needed
SUB RATE	Apply Rate	Payoff = Bill Amount / Current = Amount Due	Get Consumption From Master Bill Segment	N/A - rated sub service agreements are cancelled when their master is canceled.
SUB BC	Billable Charge	Payoff = Bill Amount / Current = Amount Due	N/A - no consumption is needed	N/A - no consumption is needed

Just to make sure everything has been designed appropriately, we will return to our SA type samples and specify their respective bill segment types:

Div. SA Type	Calculation Algorithm	FT Algorithm	Consumption Algorithm	Auto Cancel Algorithm	Bill Segment Type
CA/G-RES	Apply Rate To Usage Request	Payoff = Bill Amount / Current = Amount Due	Get Consumption Using Usage Request	NA - can't estimate consumption.	BD RATED
CA/G-COM	Apply Rate To Usage Request	Payoff = Bill Amount / Current = Amount Due	Get Consumption Using Usage Request	NA - can't estimate consumption.	BD RATED
CA/E-RES	Apply Rate To Usage Request	Payoff = Bill Amount / Current = Amount Due	Get Consumption Using Usage Request	NA - can't estimate consumption.	BD RATED
CA/E-COM	Apply Rate To Usage Request	Payoff = Bill Amount / Current = Amount Due	Get Consumption Using Usage Request	NA - can't estimate consumption.	BD RATED
CA/G-IND	Apply Rate To Usage Request	Payoff = Bill Amount / Current = Amount Due	Get Consumption Using Usage Request	NA - can't estimate consumption.	BD RATED
CA/CABLE	Apply Rate To Usage Request	Payoff = Bill Amount / Current = Amount Due	Get Consumption Using Usage Request	NA - can't estimate consumption.	BD RATED

CA/E-COY	Apply Rate To Usage Request	Payoff = 0 / Current = Bill Amount	Get Consumption Using Usage Request	N/A - no consumption is needed.	COMPUSAG
CA/WO-STD	N/A - non billable	N/A - non billable	N/A - non billable	N/A - non billable	
CA/WO-LIA	N/A - non billable	N/A - non billable	N/A - non billable	N/A - non billable	
CA/CHARITY	Recurring Charge	Payoff = 0 / Current = Bill Amount	N/A - no consumption is needed	N/A - no consumption is needed	RECUR
CA/PA-REGU	Recurring Charge With Auto Stop	Payoff = 0 / Current = Bill Amount	N/A - no consumption is needed	N/A - no consumption is needed	RECUR-AS
CA/MERCH-I	Recurring Charge For Amount To Bill	Payoff = 0 / Current = Bill Amount	N/A - no consumption is needed	N/A - no consumption is needed	RECUR-AS
CA/DEP-I	Recurring Charge For Amount To Bill	Payoff = 0 / Current = Bill Amount	N/A - no consumption is needed	N/A - no consumption is needed	RECURATB
CA/ONETIME	Billable Charge	Payoff = Bill Amount / Current = Amount Due	N/A - no consumption is needed	N/A - no consumption is needed	BILLCHRG
CA/OVR UNDR	N/A - non billable	N/A - non billable	N/A - non billable	N/A - non billable	
CA/E-SUB ENR	Apply Rate	Payoff = Bill Amount / Current = Amount Due	Get Consumption From Master Bill Segment	N/A - rated sub service agreements are cancelled when their master is canceled.	SUB RATE
CA/E-SUB BC	Billable Charge	Payoff = Bill Amount / Current = Amount Due	N/A - no consumption is needed	N/A - no consumption is needed	SUB BC
CA/E-FIN SETTL	N/A - non billable	N/A - non billable	N/A - non billable	N/A - non billable	

And now you’re ready to set up your bill segment types.

Setting Up Bill Segment Types

To set up bill segment types, open **Admin** > **Billing** > **Bill Segment Type** > **Add.**

Description of Page

Enter an easily recognizable **Bill Segment Type** and **Description** for every type of bill segment.

For each bill segment type, define the **Create Algorithm**. The logic embedded in this algorithm creates the bill segment. Refer to Designing Your Bill Segment Types for examples.

If you haven't done so already, you must set up this algorithm in the system. To do this:

- Create a new algorithm (refer to Setting Up Algorithms).
- On this algorithm, reference an Algorithm Type that creates a bill segment in the appropriate manner. Click here to see the algorithm types available for this plug-in spot.

WARNING:
The **BS Create Algorithm** is a very important field as it controls how the system creates bill segments. There are some restrictions in respect of the values of certain fields on the SA type and the bill segment algorithm used on a SA type. Refer to Require Total Amount Switch versus Bill Segment Creation Algorithm, Allow Recurring Charge Switch versus Bill Segment Creation Algorithm, and Rate Required Switch versus Bill Segment Creation Algorithm**for more information**.

For each bill segment type, define the **Financial Algorithm**. The logic embedded in this algorithm constructs the financial transaction associated with the bill segment. Refer to Designing Your Bill Segment Types for examples.

If you haven't done so already, you must set up this algorithm in the system. To do this:

- Create a new algorithm (refer to Setting Up Algorithms).
- On this algorithm, reference an Algorithm Type that constructs the bill segment financial transaction in the appropriate manner. Click here to see the algorithm types available for this plug-in spot.

●

FASTPATH:
For more information about current and payoff amounts, refer to Current Amount versus Payoff Amount.

If the bill segment requires consumption (e.g., meter reads) to be retrieved, define the **Get Consum Algorithm**. The logic embedded in this algorithm retrieves the consumption that is billed on the bill segment. Refer to Designing Your Bill Segment Types for examples.

If you haven't done so already, you must set up this algorithm in the system. To do this:

- Create a new algorithm (refer to Setting Up Algorithms).
- On this algorithm, reference an Algorithm Type that retrieves consumption in the appropriate manner. Click here to see the algorithm types available for this plug-in spot.

The **Auto Cancel Algorithm** is used by the system when it detects that a prior bill segment contains an estimated read that meets certain conditions when a non-estimated read is used on the current bill.

➤ **NOTE:** This is typically not used when usage is requested from Oracle Utilities Meter Data Management.

Examples include:

- The prior bill segment contains a bad estimated read (by "bad" we mean that the current bill has a non-estimated reading that is less than the estimated end read on the prior bill segment).
- The prior bill segment contains an estimated read (there may be situations where you want to cancel an estimated bill segment when a non-estimated read is also higher than the end read on the prior bill segment. For example, when tiered rates may cause a customer to be penalized by consumption being charged at a higher rate).

For detecting a bad estimated read on the prior bill segment, if you haven't done so already, you must set up this algorithm in the system. To do this:

- Create a new algorithm (refer to Setting Up Algorithms).
- On this algorithm, reference an Algorithm Type that cancels bill segments that contain poorly estimated consumption. Click here to see the algorithm types available for this plug-in spot.

For detecting an estimated read on the prior bill segment, if you haven’t done so already, you must set up this algorithm in the system and configure a feature configuration option. To do this:

- Create a new algorithm for the Auto Cancel Algorithm.
- On this algorithm, reference an Algorithm Type that cancels bill segments that contain estimated consumption. Click here to see the algorithm types available for this plug-in spot (refer to Setting Up Algorithms).
- Configure the **Always Call Auto Cancel** option type on the Financial Transactions Option feature configuration and set the option type value to 'Y'. For more information about Feature Configurations, see Defining Feature Configurations.

The **Bill Segment Information Algorithm** is used by the system to format the bill segment information that appears throughout the system. If the information you'd like displayed differs for bill segment types, you must set up this algorithm in the system. To do this:

Where Used

Follow this link to open the data dictionary where you can view the tables that reference CI_BILL_SEG_TYP.

Designing and Defining Deposit Classes

If you bill for deposits, you must set up one or more deposit classes. If your company does not bill for deposits, you can skip this section.

●

FASTPATH:
We strongly recommend familiarizing yourself with the concepts described in The Big Picture Of Deposits before tackling the information in this section.

The topics in this section describe how to design and set up deposit classes.

What Do Deposit Classes Do?

A deposit class contains the business rules that govern:

- How and when deposit interest is calculated.
- How the recommended deposit amount is calculated.
- When a deposit will be automatically refunded to a customer.
- When the system will recommend a new or additional deposit.

When you link a deposit class to a SA type, you are indicating that the SA type's service agreements are governed by the deposit class' business rules.

In addition to linking a deposit class to the SA types used to bill for a deposit, you must also define a deposit class on SA types whose debt is covered by a deposit. Consider the following examples:

- Assume your company sells electricity, gas, and water; but deposits are only held only for electric service. In this situation, you'd need one deposit class - Electric - and you'd associate it with both the electric deposit SA type and the electric usage SA type(s) (the gas and water SA types would NOT reference a deposit class).
- If your company can apply a deposit to any type of debt, then you'd have just one deposit class - General Deposit. You'd link this deposit class to the deposit SA type, and to the other SA types whose debt is covered by the deposit.

➤

NOTE:
Non-cash deposits. You can also use the system to manage non-cash deposits (e.g., letters of credit, surety bonds, 3 rd party deposits). Non-cash deposits are held in respect of an account (and an account may have

> an unlimited number of non-cash deposits). Each non-cash deposit must reference a deposit class. Why? Because the system amalgamates cash and non-cash deposits when it determines if an account is holding an adequate deposit. Refer to 3rd Party Deposits for more information.

Designing Your Deposit Classes

A deposit class contains the business rules that govern:

- How and when deposit interest is calculated.
- How the recommended deposit amount is calculated.
- When a deposit will be automatically refunded to a customer.
- When the system will recommend a new or additional deposit.

You will need multiple deposit classes if any of the above rules / conditions differ for different types of customers. For example, if residential customers use a different recommended deposit algorithm as compared to commercial customers, you'd need one deposit class for residential and another for commercial.

You will need additional deposit classes if your customers can have multiple deposits where each deposit is restricted to a specific type of debt. For example, if separate deposits are held for regulated and unregulated debt (and a customer could hold a combination of regulated and unregulated debt), you'd need one deposit class for regulated debt and another for unregulated debt.

We'll design deposit classes to satisfy the needs of a theoretical company to help you understand how to design your deposit classes. The following points describe the deposit requirements of our theoretical company:

- The recommended deposit amount is 2 times the average bill (averaged over the last 12 months). This is true regardless of the type of customer or debt.
- The system should automatically refund a deposit to a customer after:
  - The deposit has been held for at least 6 months; and
  - The account's credit rating is greater than the credit rating threshold defined on the installation record or the account’s CIS Division (i.e., the credit rating is no longer considered bad)
- This is true regardless of the type of customer or debt.
- Interest is calculated every 6 months. The interest rate is defined using a bill factor (refer to Setting Up Bill Factors for more information). This is true regardless of the type of customer or debt.
- When it's time to refund a refund a deposit, all outstanding debt will be paid off first. If any moneys remain, a check should be sent to the customer for the remainder. This is true regardless of the type of customer or debt.
- A customer could have both regulated and unregulated debt under a single account. When this happens, separate deposits will be held for each type of debt (where the regulated deposit can only be used to satisfy regulated debt and the unregulated deposit can only be used to satisfy unregulated debt).

You'd need the following deposit classes to satisfy the above requirement:

Deposit Class	Recommended Amount Rule	Auto Refund Condition	Interest Rules	Deposit Refund Method
Regulated	2 x Average Bill	Held for 6 months and credit rating is good	Simple interest every 6 months	Apply to outstanding debt first, refund remainder with a check

Unregulated	2 x Average Bill	Held for 6 months and credit rating is good	Simple interest every 6 months	Apply to outstanding debt first, refund remainder with a check

You may wonder why two deposit classes are needed when the rules are the same for both? Well, besides defining the applicable business rules for a deposit service agreement, a deposit class is defined on the SA types whose debt is covered by the deposit class' deposit. So, if you have two different types of debt where each type of debt can have its own deposit, you'd need at two deposit classes. Each deposit class would be associated with the service agreements that are being secured by the deposit.

Refer to Setting Up Deposit Classes for a description of the various algorithms defined in respect of a deposit class.

Setting Up Deposit Classes

In the previous section, Designing Your Deposit Classes, we presented a case study that illustrated a mythical organization's deposit classes. In this section, we explain how to maintain your Deposit Classes.

Deposit Class - Main

To set up deposit classes, select **Admin** > **Credit & Collection** > **Deposit Class** > **Add.**

Description of Page

Enter an easily recognizable **Deposit Class** and **Description**.

Use **Refund Description on Bill** to define the information that appears on the bill segment produced when it's time to refund the customer's deposit.

The remaining information on this page is used by the various deposit-oriented processes.

Refer to Deposit Class - Good Customer for information about the **Good Customer Algorithm**.

Refer to Deposit Class - Recommendation for information about the **Recommendation Algorithm** and **Review Tolerance Percentage**.

Refer to Deposit Class - Refund Method for information about the **Refund Method Algorithm**.

Refer to Deposit Class - Refund Criteria for information about the **Refund Criteria Algorithm**.

Refer to Deposit Class - Refund Interest for information about the **Interest Refund Algorithm** and **Months Between Interest Refund**.

Refer to Deposit Class - Review Method for information about the **Review Method Algorithm**.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference CI_DEP_CL.

Deposit Class - Good Customer

On Deposit Class - Main you must define the **Good Customer Algorithm** used by the system when it determines if a customer is considered good (the system recommends new / additional deposits for bad customers). Refer to Deposit Review for a description of the background process that reviews customers for adequate deposits.

If you haven't done so already, you must set up this algorithm in the system. To do this:

- Create a new algorithm (refer to Setting Up Algorithms).

- On this algorithm, reference an Algorithm Type that determines if a customer is considered good. Click here to see the algorithm types available for this plug-in spot.

Deposit Class - Recommendation

On Deposit Class - Main you must define the **Recommendation Algorithm** used by the system when it calculates a customer's recommended deposit amount.

If you haven't done so already, you must set up this algorithm in the system. To do this:

- Create a new algorithm (refer to Setting Up Algorithms).
- On this algorithm, reference an Algorithm Type that recommends deposits. Click here to see the algorithm types available for this plug-in spot.

The system uses the **Review Tolerance Percentage** to prevent the recommendation of small deposits by the Deposit Review background process. For example, if this field contains 10(%), the system would only recommend an additional deposit if the recommended amount was more than 10% of the existing deposit.

Deposit Class - Refund Method

On Deposit Class - Main you must define the **Refund Method Algorithm** used by the system when it refunds a deposit to the customer.

If you haven't done so already, you must set up this algorithm in the system. To do this:

- Create a new algorithm (refer to Setting Up Algorithms).
- On this algorithm, reference an Algorithm Type that refunds a deposit to a customer. Click here to see the algorithm types available for this plug-in spot.

Deposit Class - Refund Criteria

On Deposit Class - Main you must define the **Refund Criteria Algorithm** used by the system when it determines if it should automatically refund a deposit to a customer. Refer to Deposit Review for a description of the background process that reviews deposits for refunds.

If you haven't done so already, you must set up this algorithm in the system. To do this:

- Create a new algorithm (refer to Setting Up Algorithms).
- On this algorithm, reference an Algorithm Type that determines if a customer qualifies for a deposit refund. Click here to see the algorithm types available for this plug-in spot.

Deposit Class - Refund Interest

On Deposit Class - Main you must define the **Interest Refund Algorithm** to define how the system calculates interest and how it refunds the interest to the customer.

If you haven't done so already, you must set up this algorithm in the system. To do this:

- Create a new algorithm (refer to Setting Up Algorithms).

- On this algorithm, reference an Algorithm Type that calculates interest on a deposit. Click here to see the algorithm types available for this plug-in spot.

Interest will be automatically calculated every X months where X is defined in **Months Between Interest Refund**. Refer to Deposit Interest for a description of the background process that calculates interest on deposits. Also note that interest is calculated when a deposit service agreement is stopped.

Deposit Class - Review Method

On Deposit Class - Main you must define the **Review Method Algorithm** used by the system to determine what action to take if the system recommends a deposit (or additional deposit) amount for an account. Refer to Review Deposits for a description of the background process that reviews deposits for refunds. The algorithm supplied with the base product highlights new deposits and deposit amounts on the Deposit Review page.

If you haven't done so already, you must set up this algorithm in the system. To do this:

- Create a new algorithm (refer to Setting Up Algorithms).
- On this algorithm, reference an Algorithm Type that determines the review method the system uses if it recommends a deposit or additional deposit be applied to an account. Click here to see the algorithm types available for this plug-in spot.

Setting Up Non-Cash Deposit Types

Non-cash deposit types are used to indicate the type of monetary instrument used for non-cash deposits. Refer to Non-Cash Deposits for more information.

To define your non-cash deposit types, select **Admin** > **Credit & Collection** > **Non-Cash Deposit Type.**

Description of Page

To modify a non-cash deposit type, move to a field and change its value.

To add a new non-cash deposit type, insert a row, then fill in the information for each field. The following fields display:

Non-Cash Deposit Type The unique identifier of the non-cash deposit type.

Description The description of the non-cash deposit type.

Review Before Expiration This switch indicates if the system will create a To Do entry when non-cash deposits of this type are close to expiration.

Third Party Deposit This switch indicates if the system requires a reference to a 3rd party's deposit service agreement for this type of non-cash deposit. Refer to 3rd Party Deposits for more information.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference CI_NCD_TYPE.

Setting Up Payment Segment Types

Every service agreement references a service agreement (SA) type. Amongst other things, the SA type references a payment segment type. The payment segment type controls how payment segments and their related financial transactions are created. To set up payment segment types, open **Admin** > **Payment Segment Type.**

Description of Page

Enter an easily recognizable **Payment Segment Type** and **Description** for every type of payment segment.

- **FASTPATH:**
  For more information about the source of the distribution codes on financial transactions, see The Source Of GL Accounts On Financial Transactions.

For each payment segment type, define the **Payment Segment Fin Algorithm**. The logic embedded in this algorithm constructs the actual financial transaction associated with the payment segment. Refer to Examples of Common Payment Segment Types for examples of how algorithms are used on common payment segment types.

If you haven't done so already, you must set up this algorithm in the system. To do this:

- Create a new algorithm (refer to Setting Up Algorithms).
- On this algorithm, reference an Algorithm Type that constructs the payment segment financial transaction in the appropriate manner. Click here to see the algorithm types available for this plug-in spot.

- **FASTPATH:**
  For more information about current and payoff amount, see Current Amount versus Payoff Amount.

Examples of Common Payment Segment Types

The following table shows several classic payment segment types used by many organizations:

Payment Segment Type	Payment Segment Financial Transaction Algorithm
Normal payment (if you practice accrual accounting). Refer to Accrual versus Cash Accounting for more information.	Payoff = Pay Amount / Current = Pay Amount (no cash accounting)
Normal payment (if you practice cash accounting). Refer to Accrual versus Cash Accounting for more information.	Payoff = Pay Amount / Current = Pay Amount (plus Cash Accounting)
Charity payment	Payoff =0 / Current = Pay Amount (the GL is affected)
Non-CIS Payments (When the FT is created, the distribution code and GL account to credit is retrieved from the pay). Refer to Non CIS Payments for more information	Payoff = Pay Amount / Current = Pay Amount (no cash accounting)

Where Used

Follow this link to open the data dictionary where you can view the tables that reference CI_PAY_SEG_TYPE.

Designing And Defining Adjustment Types

A service agreement's debt may be changed with an adjustment. Every adjustment must reference an adjustment type. The adjustment type contains a great deal of information that is defaulted onto the adjustment, including whether the adjustment amount is calculated. It also controls many business processes associated with the adjustment. The topics in this section describe how to design and set up adjustment types.

What Do Adjustment Types Do?

An adjustment type contains the business rules that govern how its adjustments are managed by the system. Please refer to The Big Picture Of Adjustments for a complete description of how adjustment types impact the lifecycle of adjustments.

Setting Up Adjustment Types

The topics in this section describe how to set up adjustment types.

➤ NOTE:

When a new adjustment type is added. When you introduce a new adjustment type, you must update one or more adjustment profiles with the new adjustment type. Why? Because adjustment profiles define the adjustment types that may be levied on service agreements (adjustment profiles are defined on SA types). If you don't put the adjustment type on an adjustment profile, the adjustment type can't be used on any adjustment.

Adjustment Type - Main

To set up adjustment types, open **Admin** > **Financial** > **Adjustment Type** > **Add.**

Description of Page

Enter a unique **Adjustment Type** ID and **Description** for the adjustment type.

If your implementation requires restricting the list of adjustments types that users are able to select by CIS Division, specify a **CIS Division** on the adjustment type.

The **AdjustmentAmount Type** indicates whether the adjustment about is calculated or not. Select Calculated Amount when you want to use a rate to perform calculations to generate the adjustment amount otherwise select Non-Calculated Amount. Refer to Setting Up Calculated Adjustment Types for more information about calculated adjustments.

Enter the **Distribution Code** that references the GL account associated with the adjustment. For example, if this adjustment type is used to levy a charge for a bad check, the distribution code would reference the revenue account to which you associate such revenue. Note, the offsetting distribution code is kept on the SA type.

➤ NOTE:

Distribution Code for Calculated Adjustments. Depending on the algorithm used for the calculated adjustment, the distribution code may come from the adjustment type or the calculation lines of the algorithm. If the adjustment's calculation algorithm gets the distribution code from the calculation lines, you do not need to specify a distribution code on the adjustment type.

● FASTPATH:

For more information about the source of the distribution codes on financial transactions, see The Source Of GL Accounts On Financial Transactions.

Enter the **Currency Code** for adjustments of this type.

Turn on **Sync. Current Amount** if adjustments of this type exist to force a service agreement's current balance to equal its payoff balance. These types of adjustments are issued before a service agreement's funds are transferred to a write-off service agreement. If this switch is on, choose an **Adjustment Fin Algorithm** that does not impact payoff balance or the GL, but does affect the SA's current balance (refer to ADJT-CA for an example of such an algorithm).

Enter a **Default Amount** if an amount should be defaulted onto adjustments of this type.

- **FASTPATH:**
  For more information about current and payoff amounts, refer to Current Amount versus Payoff Amount.

If the AP Adjustment should be recorded in respect of the customer's 1099 amounts, indicate the **A/P 1099 Flag**. This would typically be used on the adjustment used to credit the deposit service agreement with accrued interest. The values of this field are Interest and Miscellaneous. This type of adjustment would also have an **A/P Request Type Code** selected, as 1099 reporting is handled in A/P.

Turn on **Print By Default** if information about adjustments of this type should print on the account's next bill.

Choose an **A/P Request Type Code** if this adjustment is interfaced to accounts payable (i.e., it's used to send a refund check to a customer). Refer to A/P Check Request for more information.

The **Adjustment Freeze Option** defines when adjustments can be frozen and therefore when a service agreement's balance and the general ledger are affected by an adjustment. Refer to Preventing SA Balances And The GL From Being Impacted Until Bill Completion to understand the significance of this option. Also note, if the installation option's Bill Segment Freeze Option is Freeze At Will, this field is defaulted to Freeze At Will and cannot be changed.

WARNING:
Adjustment types for adjustments created during bill completion (e.g., by a bill completion algorithm) must have their adjustment freeze option set to Freeze At Will. Otherwise (i.e., if the option is Freeze At Bill Completion) they will not be frozen until a subsequent bill is completed.

If adjustments of this type require approval, define an **Approval Profile**. For more information, refer to The Big Picture of Adjustment Approvals.

Enter the verbiage to appear on the printed bill in **Description on Bill**.

Use the characteristics collection to define a **Characteristic Type** and **Characteristic Value** common to all adjustments of this type. These can be used for reporting purposes or in your algorithms.

Adjustment Type - Adjustment Characteristics

To define characteristics that can be defined for adjustments of a given type, open **Admin** > **Financial** > **Adjustment Type** > **Search** and navigate to the **Adjustment Characteristics** page.

Description of Page

Use the **Adjustment Characteristics** collection to define characteristics that can be defined for adjustments of a given type. Turn on the **Required** switch if the **Characteristic Type** must be defined on adjustments of a given type. Enter a

Characteristic Value to use as the default for a given **Characteristic Type** when the **Default** switch is turned on. Use **Sequence** to control the order in which characteristics are defaulted.

Adjustment Type - Algorithms

To define algorithms for adjustments, open **Admin** > **Financial** > **Adjustment Type** > **Search** and navigate to the **Algorithms** page.

Description of Page

The grid contains **Algorithms** that control important adjustment functions. If you haven't already done so, you must set up the appropriate algorithms in your 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.

The following table describes each **System Event**.

System Event	Optional / Required	Description
Adjustment Cancellation	Optional	When an adjustment is canceled an algorithm of this type may be called to do additional work. Refer to The Lifecycle Of An Adjustment for more information about canceling an adjustment. Click here to see the algorithm types available for this system event.
Adjustment Freeze	Optional	When an adjustment is frozen an algorithm of this type may be called to do additional work. Refer to The Lifecycle Of An Adjustment for more information about freezing an adjustment. Click here to see the algorithm types available for this system event.
Adjustment Information	Optional	We use the term "Adjustment information" to describe the basic information that appears throughout the system to describe an adjustment. The data that appears in "Adjustment information" is constructed using this algorithm. Plug an algorithm into this spot to override the "Adjustment information" algorithm on installation options or the system default "Adjustment information" if no such algorithm is defined on installation options. Click here to see the algorithm types available for this system event.

Adj. Financial Transaction	Required	Algorithms of this type are used to construct the actual financial transaction associated with the adjustment. The financial transaction controls the adjustment's affect on the service agreement's payoff and current balances. It also constructs the information that is eventually interfaced to your general ledger. Refer to Examples of Common Adjustment Types for examples of how algorithms are used on common adjustment types. Click here to see the algorithm types available for this system event.
Default Adjustment Amount	Optional	Algorithms of this type are used to default the adjustment amount. Refer to Default the Adjustment Amount for more information. Click here to see the algorithm types available for this system event.
Determine SA	Optional	Algorithms of this type are used to find a service agreement for which the adjustment can be posted. This plug-in is used particularly during adjustment upload when a staging record does not identify the SA ID. Refer to Interfacing Adjustments From External Sources for more information. Click here to see the algorithm types available for this system event.
Resolve Suspense	Optional	Algorithms of this type are used to automatically resolve adjustments that are in suspense. Refer to Suspense Adjustments for more information Click here to see the algorithm types available for this system event.
Generate Adjustment	Optional	Algorithms of this type are used to calculate the adjustment amount if an adjustment type indicates that the adjustment amount is calculated. Refer to Setting Up Calculated Adjustment Types for more information. Click here to see the algorithm types available for this system event.
Validate Adjustment	Optional	Algorithms of this type are used to validate information for the adjustment after it is generated. Click here to see the algorithm types available for this system event.

Setting Up Calculated Adjustment Types

You can use an algorithm to calculate an adjustment amount for example if you need to calculate tax on a base amount or calculate a non-sufficient funds charge based on the customer's credit rating. Because the base package algorithm calculates adjustment amounts by calling the rate application, calculated adjustments are sometimes referred to as ratable adjustments.

> **NOTE:**
> **Ratable Adjustments Appear Deceptively Simple.** But, they are not. Calculated adjustments that use the base package algorithm have all the power and flexibility (and complexity) of the rate application. Anything that you can do with a rate can be applied to a calculated adjustment. For examples that illustrate the flexibility of the rate application (and therefore calculated adjustments), refer to Rate Examples.

Adjustment types that indicate they are calculated have a generate adjustment algorithm. The base package algorithm defines the rate schedule used to calculate the adjustment as well as any UOM, TOU or SQI parameters.

To set up calculated adjustment types using the base package generate adjustment algorithm type:

- Define the rate that performs the calculations, including the rate schedule, calculation groups, and calculation rules. Refer to Rates for information.

> **NOTE:**
> If you create your own Generate Adjustment algorithm type, you may not need to set up a rate that performs the calculations. It depends on the needs of your algorithm type.

- Create a Generate Adjustment algorithm (refer to Setting Up Algorithms) that references the base package algorithm type that generates calculated adjustments (see the table above).
- If you want the generation algorithm's calculation lines to provide the distribution codes when the adjustment is posted to the GL, create an Adjustment Financial Transaction algorithm (refer to Setting Up Algorithms) that references an algorithm type that creates the adjustment's financial transactions using the calculation lines. A parameter of the adjustment financial transaction algorithm determines whether the distribution codes are taken from the adjustment type (AT) or calculation lines (CL). The system comes supplied with several sample algorithm types that create adjustment financial transactions.
- Create an adjustment type where the **Adjustment Amount Type** is Calculated Amount, the Generate Adjustment event references the generation algorithm created above, and the Adj. Financial Transaction event references the adjustment financial transaction algorithm created above.

Examples of Common Adjustment Types

The following table shows several classic adjustment types used by many organizations:

Adjustment Type	Typical Distribution Code	Default Amount	Financial Transaction Algorithm	Print On Bill	A/P Adjust-ment	1099
NSF	NSF revenue	Your NSF fee	Payoff = Current = Adj. Amt Refer to ADJT-NM for an example of such an algorithm type	Yes	No	No

LPC	Late payment charge revenue	N/A	Payoff = Current = Adj. Amt Refer to ADJT-NM for an example of such an algorithm type	Yes	No	No
CONNECT	Connection charge revenue	Your connection charge	Payoff = Current = Adj. Amt Refer to ADJT-NM for an example of such an algorithm type	Yes	No	No
CUSTREL	Customer relationship expense	N/A	Payoff = Current = Adj. Amt Refer to ADJT-NM for an example of such an algorithm type	Yes	No	No
ADDCHARG	Misc Revenue	N/A	Payoff = Current = Adj. Amt Refer to ADJT-NM for an example of such an algorithm type	Yes	No	No
XFER	Balance transfer clearing	N/A	Payoff = Current = Adj. Amt Refer to ADJT-NM for an example of such an algorithm type	Yes	No	No
WO SYNC	N/A	N/A	Payoff = 0 / Current = Adj. Amt Refer to ADJT-CA for an example of such an algorithm type	No	No	No
REFUNDAP	A/P clearing	N/A	Payoff = Current = Adj. Amt Refer to ADJT-NM for an example of such an algorithm type	Yes	Yes	No
DPA FIX	N/A	N/A	Payoff = 0 / Current = Adj. Amt Refer to ADJT-CA for an	Yes	No	No

			example of such an algorithm type			
CHARITFX	N/A	N/A	Payoff = 0 / Current = Adj. Amt Refer to ADJT-CA for an example of such an algorithm type	Yes	No	No
BUDG ON	N/A	N/A	Payoff = 0 / Current = Adj. Amt Refer to ADJT-CA for an example of such an algorithm type	Yes	No	No
BUDG OFF	N/A	N/A	Payoff = 0 / Current = Adj. Amt Refer to ADJT-CA for an example of such an algorithm type	Yes	No	No
BUDG FIX	N/A	N/A	Payoff = 0 / Current = Adj. Amt Refer to ADJT-CA for an example of such an algorithm type	Yes	No	No
DEPOSREF	N/A	N/A	Payoff = 0 / Current = Adj. Amt Refer to ADJT-CA for an example of such an algorithm type	Yes	No	No
DEPOSINT	Interest expense	N/A	Payoff = Current = Adj. Amt. Refer to ADJT-NM for an example of such an algorithm type. Or Payoff Amt = Adj Amt / Current Amt = 0. Refer to ADJT-TA for an example of	Yes	No	Yes

			such an algorithm type. Use the first method if you want to have the interest reflected as a credit balance on the customer's bill. Use the second method if you roll the interest amount into the customer's existing deposit on hand.			
DEPFIXCR	N/A	N/A	Payoff = 0 / Current = Adj. Amt Refer to ADJT-CA for an example of such an algorithm type	No	No	No

Where Used

Follow this link to open the data dictionary where you can view the tables that reference CI_ADJ_TYPE.

Setting Up Adjustment Type Profiles

Adjustment type profiles categorize your adjustment types into logical groups. When you link a profile to a SA type, you limit the type of adjustments to be linked to the SA type's service agreements. The creation of adjustment profiles and their linkage to SA types prevents inappropriate adjustments from being linked to your service agreements. More than one adjustment type profile may be linked to a SA type.

For example, you can create an adjustment type profile called Miscellaneous Fees and link to it the miscellaneous fee adjustment types. Then, you would link this profile to those SA types that are allowed to levy such fees.

NOTE:
Bottom line. An adjustment can only be linked to a service agreement if its adjustment type is part of an adjustment type profile that is valid for the service agreement's SA type. If an adjustment type is not linked to a profile, it could never be levied.

To set up adjustment type profiles, open **Admin** > **Financial** > **Adjustment Type Profile** > **Add.**

Description of Page

Enter a unique **Adjustment Type Profile** and **Description** for the adjustment type profile.

Indicate the **Adjustment Types** that are part of the profile.

Examples Of Common Adjustment Profiles

The following table shows several classic adjustment profiles used by many organizations (we've displayed some attributes from the adjustment type in the following table to help make it more understandable):

Adjustment Profile	Adjustment Type	Typical Distribution Code	Default Amount	Financial Transaction Algorithm
FEES	NSF	NSF revenue	Your NSF fee	Payoff Amt = Adj Amt / Current Amt = Adj Amt
	LPC	Late payment charge revenue	Your LPC	Payoff Amt = Adj Amt / Current Amt = Adj Amt
	CONNECT	Connection charge revenue	Your connection charge	Payoff Amt = Adj Amt / Current Amt = Adj Amt
MISCEXP	CUSTREL	Customer relationship expense	N/A	Payoff Amt = Adj Amt / Current Amt = Adj Amt
XFER	TRANSBAL	Balance transfer clearing	N/A	Payoff Amt = Adj Amt / Current Amt = Adj Amt
REFUND	REFUND	A/P clearing	N/A	Payoff Amt = Adj Amt / Current Amt = Adj Amt
DPA	ADJCURR	N/A	N/A	Payoff Amt = 0 / Current Amt = Adj Amt
	SYNCCURR	N/A	N/A	Payoff Amt = 0 / Current Amt = Adj Amt
CHARITY	CHAR FIX	N/A	N/A	Payoff Amt = 0 / Current Amt = Adj Amt
BUDGET	BUDG ON	N/A	N/A	Payoff Amt = 0 / Current Amt = Adj Amt
	BUDG OFF	N/A	N/A	Payoff Amt = 0 / Current Amt = Adj Amt
	BUDG FIX	N/A	N/A	Payoff Amt = 0 / Current Amt = Adj Amt
	FIX PAY	Customer relationship expense	N/A	Payoff Amt = Adj Amt / Current Amt = 0
DEPOSIT	DEPOSBILL	N/A	Your standard deposit amount	Payoff Amt = 0 / Current Amt = Adj Amt
	DEPOSINT	Interest expense	N/A	Payoff Amt = Adj Amt / Current Amt = Adj Amt
	SYNCCURR	N/A	N/A	Payoff Amt = 0 / Current Amt = Adj Amt
	ADJCUR	N/A	N/A	Payoff Amt = 0 / Current Amt = Adj Amt

Where Used

Follow this link to open the data dictionary where you can view the tables that reference CI_ADJ_TYP_PROF.

The Big Picture Of Adjustment Approval

Some implementations require adjustments to be approved by one or more managers before they impact a customer's debt and the general ledger. For example, an adjustment used to rebate a credit balance may require managerial approval before the rebate is sent to the customer. The topics in this section describe how to set up the system to support the approval of adjustments.

Approval Is Controlled By Approval Profiles

An approval profile contains the rules that define if and how an adjustment is approved. If an adjustment type does not reference an approval profile, the related adjustments do not require third-party approval before they impact a customer's debt. If an adjustment type references an approval profile, the approval profile's approval hierarchy defines if the adjustment requires approval and who the authorized approvers are. For example, an approval profile can be configured with the following approval hierarchy:

- Adjustments < $0 require approval by the "credit approvers role"
- Adjustments >= $0 and <= $10 do not require approval
- Adjustments > $10 and <= $100 require the approval of a user that belongs to the "level 1 approvers role"
- Adjustments > $100 require two levels of approval: first a user that belongs to the "level 1 approvers role" must approve the adjustment; afterwards, the adjustment must be approved by a user that belongs to the "level 2 approvers role"

> **NOTE:**
> **Transfer adjustments.** The term "transfer adjustment" refers to two adjustments that are used to transfer moneys between two service agreements. The adjustment with the positive amount is considered to be the debit adjustment; the other adjustment is considered the credit adjustment. When a transfer adjustment requires approval, only one of the adjustments needs to be approved. You control whether the debit side or the credit side of a transfer adjustment is used to control the approval process when you set up the approval profile.

Approval Profiles Can Be Linked To Multiple Adjustment Types

Approval hierarchies are frequently the same for many adjustment types. The system allows an approval profile to be linked to multiple adjustment types to simplify the definition and maintenance of the rules over time.

Adjustments Created In Batch Are Not Approved

The system assumes that no approval is necessary for adjustments created by batch processes even those whose adjustment type references an approval profile.

Approval Inserts A Step Into An Adjustment's Lifecycle

The Lifecycle Of An Adjustment explains how an adjustment is transitioned from the Freezable state to the Frozen state when it should impact the general ledger and the customer's balance. If an adjustment's adjustment type references an approval profile, the user cannot freeze the adjustment directly. Rather, the user must submit the adjustment for approval when it's ready and only when the last applicable approver approves the adjustment will it become Frozen.

NOTE:
Freeze during bill completion. You can configure the system to only freeze certain types of adjustments when the next bill is completed for the adjustment's account. When the last approver approves such adjustments, they remain in the Freezable. When the next bill is completed for the account, these adjustment become Frozen. Such adjustments that have not been approved at the time of bill completion will remain in the Freezable state. Refer to Preventing SA Balances And The GL From Being Impacted Until Bill Completion for more information.

Approval Requests Manage And Audit The Approval Process

Users submit an adjustment for approval using a dedicated button on the Adjustment page. When an adjustment is submitted for approval, the system creates an "approval request". The approval request determines if the adjustment requires approval and, if so, the list of approvers. If the adjustment does not require approval, the approval request is updated to indicate such and the adjustment is Frozen immediately (if freezing is allowed prior to bill completion). If the adjustment requires approval, the approval request's state becomes Approval In Progress and the approver(s) are notified.

NOTE:
Approval submission logic is customizable. The previous paragraph describes how the base-package works when an adjustment is submitted for approval. This logic resides in an algorithm that's plugged in on the C1-AdjustmentApprovalProfile business object in the Determine Approval Requirements system event. Your implementation can change this logic by developing a new algorithm and plugging it into this business object. If your logic is meant to supersede the base-package algorithm, remember to inactivate the base-package algorithm by adding an appropriate inactivation option to this business object.

To Do Entries Are Created To Notify Approvers

When an approval request detects an adjustment requires approval, it notifies the first approver by creating a To Do entry. The To Do entry is created using the To Do type and To Do role defined on the approval profile. All users who belong to the approving To Do role can see the entry. When a user drills down on an adjustment approval To Do entry, the Adjustments - Approval portal is opened. This portal contains summary information about the adjustment and the approval history of the adjustment. This portal is also where the user approves or rejects the adjustment.

When the first user in the To Do role approves an adjustment, the To Do entry is Completed and the approval request's audit log is updated. If there are no higher levels of approval required, the adjustment is Frozen (if freezing is allowed prior to bill completion) and the approval request is moved to the Approved state. If there are higher levels of approval required, a new To Do entry is created to the next To Do role in the approval hierarchy.

NOTE:

> **To Do entries can create email.** A To Do entry can be configured to create an email message for every user in the To Do role to inform the user(s) of new adjustments requiring their attention. Refer to To Do Entries May Be Routed Out Of The System for the details.

Monitoring and Escalating Approval Requests

The base-package is supplied with an algorithm that your implementation can use to monitor approval requests that have been waiting too long for approval. This algorithm can complete the current To Do entry and create a new one for a different role when the timeout threshold defined on the algorithm's parameters is exceeded. If you've configured the system to send email for approval, this algorithm can also send x reminder emails (where x is defined on the algorithm's parameters) before the approval request is escalated to the new To Do role. Refer to C1-APR-TMOUT for more information about this algorithm. If you plan to enable this functionality, plug-in your configured algorithm on the Approval In Progress state on the C1-AdjustmentApprovalRequest business object.

Rejecting Deletes The Adjustment

When an adjustment is being approved, anyone with access to the adjustment can reject it by using the Adjustments - Approval portal. Users other than the current approver are allowed to reject an adjustment to allow an "in process" an adjustment to be withdrawn.

When an adjustment is rejected, the following takes place:

- The user is prompted for a reject reason.
- The approval request's audit log is updated with the reject reason and the approval request is moved to the Rejected state.
- The adjustment is deleted.

Designing Your Approval Profiles

The following points describe a recommended design process:

- Create logical groups of adjustment types where each group has the same monetary hierarchy and approvers. An approval profile will be required for each of these groups.
- The number of To Do types (if any) that need to be created is dependent on how the adjustment approval To Do entries should be organized on To Do lists. For example, if all approval request To Do entries can appear in the same To Do list, you can use the base-package adjustment approval To Do type. However, if your organization prefers each approval profile's To Do entries to appear in a distinct To Do list, a separate To Do type will be needed for each list. Note that the base-package is supplied with a To Do type called **C1-ADAPP** that should be used as the basis for any new approval request To Do type.
- The number of To Do roles is dependent on who approves your adjustments. At a minimum, you will require a separate To Do role for each level in your approval profiles. Remember that every user in a To Do role will see its entries (and receive email if you've configured the system to do such).
- Refer to Monitoring and Escalating Approval Requests for how to configure the system to escalate approval requests that have been waiting too long.
- If your implementation requires email notification when an adjustment requires approval, the following setup is required:
  - Set up an outbound message type, external system, and message sender. Refer to To Do Entries May Be Routed Out Of The System for the details.

- Every To Do type referenced on your approval profiles should be configured as follows:

- Define the F1-TDEER batch process as the To Do type's routing process
- Set up an algorithm that references the C1-ADJAREQEM algorithm type and plug it in the External Routing system event.

Exploring Adjustment Approval Data Relationships

Use the following links to open the application viewer where the physical tables and data relationships behind the approval functionality are documented:

- Click C1-APPR PROF to view the approval profile maintenance object's tables.
- Click C1-APPR REQ to view the approval request maintenance object's tables.

Implementing Other Approval Paradigms

The above sections describe how the base-package adjustment approval process works. Because adjustment approval has been implemented using the C1-AdjustmentApprovalProfile and the C1-AdjustmentApprovalRequest business objects, your implementation can add additional business rules and change the approval user interface as required. Alternatively, if your implementation has a radically different approval process, you can create a different business objects with their own business rules. To learn how to do this, please enroll in the Configuration Tools training class.

Setting Up Approval Profiles

Approval profiles contain the rules that control how adjustments are approved. To set up an approval profile, open **Admin** > **Financial** > **Approval Profile** > **Add**.

- **FASTPATH:**
  Refer to The Big Picture Of Adjustment Approval for a detailed description of how approval profiles govern the adjustment approval process.

The topics in this section describe the base-package zones that appear on the Approval Profile portal.

Approval Profile List

The Approval Profile List zone lists every approval profile. The following functions are available:

- Click a broadcast button to open other zones that contain more information about the adjacent approval profile.
- Click the Add link in the zone's title bar to add a new approval profile.

Approval Profile

The Approval Profile zone contains display-only information about an approval profile. This zone appears when an approval profile has been broadcast from the Approval Profile List zone or if this portal is opened via a drill down from another page. The following functions are available:

- Click the **Edit** button to start a business process that updates the approval profile.
- Click the **Delete** button to start a business process that deletes the approval profile.
- Click the **Duplicate** button to start a business process that duplicates the approval profile.

Please see the zone's help text for information about this zone's fields.

Approval Profile's Adjustment Types

The Approval Profile's Adjustment Types zone lists every adjustment type that is governed by this approval profile. This zone appears when there is at least one adjustment type governed by the approval profile displayed in the Approval Profile zone:

To add an adjustment type to this list:

- Navigate to the Adjustment Type page and display the desired adjustment type.
- Specify the governing approval profile and update the adjustment type.

To remove an adjustment type from this list:

- Navigate to the Adjustment Type page and display the desired adjustment type.
- Change or remove its approval profile and update the adjustment type.

Designing and Defining Budget Plans

If you allow your customers to pay a budget amount each month (as opposed to their actual bill amount), you must set up one or more budget plans. If your company does not offer budget billing options, you can skip this section.

The topics in this section describe how to design and set up budget plans.

The Financial Impact Of Budget Plans

The only difference between a customer who participates in budget billing and one who doesn't is that budget billing customer have bill segments where payoff amount differs from current amount. Why? Because the payoff amount is the actual amount of the bill. The current amount is the amount the customer is expected to pay (i.e., their budget amount).

Let's run through an example of a customer on a budget to illustrate a service agreement where these two balances are not the same. The values in the payoff balance and current balance columns reflect the amount due after the financial transaction has been applied:

Date	Financial Transaction	Payoff Balance	Current Balance
1-Jan-99	Bill: $125, Budget $150	125	150
15-Jan-99	Payment: $150	-25	0
2-Feb-99	Bill: $175, Budget $150	150	150
14-Feb-99	Payment: $150	0	0

3-Mar-99	Bill: $200, Budget $150	200	150
15-Mar-99	Payment: $150	50	0

FASTPATH:
For more information about current and payoff amounts, refer to Current Amount versus Payoff Amount.

What Do Budget Plans Do?

A budget plan contains the business rules that govern:

- How the recommended budget amount is calculated.
- When and how a customer on an ongoing budget plan will have their budget amount periodically trued up.
- The conditions under which the system will highlight an existing budget amount as being anomalous with the customer's current use patterns.

You may have different budget plans for different customer segments. For example, customers with large bills may have their budget amount recalculated every month, whereas small customers may have their budget amount only recalculated annually. You define which budget plans govern a customer's bills via a **budget plan on the customers' accounts**. An account's initial budget plan is defaulted from its customer class. You may override an account's budget plan at will.

Designing Your Budget Plans

FASTPATH:
Refer to Budget Billing for background information about budget billing.

A budget plan contains the business rules that govern:

- How the recommended budget amount is calculated.
- When and how a customer on an ongoing budget plan will have their budget amount periodically trued up.
- The conditions under which the system will highlight an existing budget amount as being anomalous with the customer's current use patterns.

You will need multiple budget plans if any of the above rules / conditions differ for different types of customers. For example, if residential customers use a different recommended budget algorithm as compared to commercial customers, you'd need one budget plan for residential and another for commercial.

We'll design budget plans to satisfy the needs of a theoretical company to help you understand how to design your budget plans. The following points describe the budget requirements of our theoretical company:

- The recommended budget amount is the last year's real bill amounts plus any existing debit/credit balance divided by 12. This is true regardless of the type of customer.
- The frequency of budget true up is monthly for commercial customers and annually for residential customers.

- The system should highlight when a residential customer's budget is more than 30% out of whack with what their budget amount would be if it was recalculated.
- The system should highlight when a commercial customer's budget is more than 20% out of whack with what their budget amount would be if it was recalculated.

You'd need the following budget plans to satisfy the above requirement:

Budget plan	Recommended Amount Algorithm	True Up Algorithm	Monitor Algorithm
Residential	Average Bill	True up every 12 months	Highlight when more than 30% out
Commercial	Average Bill	True up every month	Highlight when more than 20% out

Refer to the Page Controls under Setting Up Budget Plans for a description of the various algorithms defined in respect of a budget plan.

Setting Up Budget Plans

In the previous section, Designing Your Budget Plans, we presented a case study that illustrated a mythical organization's budget plans. In this section, we explain how to maintain your Budget Plans.

Budget Plan - Main

To set up budget plans, select **Admin** > **Credit & Collection** > **Budget Plan** > **Add.**

Description of Page

Enter an easily recognizable **Budget Plan** and **Description**.

If your implementation requires restricting budget plans to accounts by CIS Division, specify a **CIS Division** on the budget plan.

The remaining information on this page is used by the various budget-oriented processes.

Refer to Budget Plan - Calculation Algorithm for information about the **Calculation Algorithm**.

Refer to Budget Plan - Monitor Algorithm for information about the **Monitor Algorithm**.

Refer to Budget Plan - True Up Algorithm for information about the **True Up Algorithm** and **Months for True Up**.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference CI_BUD_PLAN.

Budget Plan - Calculation Algorithm

On Budget Plan - Main you must define the **Calculation Algorithm** used by the system when it calculates a customer's recommended budget amount.

If you haven't done so already, you must set up this algorithm in the system. To do this:

- Create a new algorithm (refer to Setting Up Algorithms).

- On this algorithm, reference an Algorithm Type that calculates recommended budget amounts. Click here to see the algorithm types available for this plug-in spot.

Budget Plan - Monitor Algorithm

On Budget Plan - Main you must define the **Monitor Algorithm** used by the Budget Monitor background process when it determines if a customer's budget plan is out-of-sync with their consumption patterns.

> ➤ **NOTE:**
> **What happens?** If the algorithm determines that a customer's budget plan is out-of-sync with its current recommended amount, an entry is added to the Budget Review page.

If you haven't done so already, you must set up this algorithm in the system. To do this:

- Create a new algorithm (refer to Setting Up Algorithms).
- On this algorithm, reference an Algorithm Type that highlights if a customer's current budget amount is out-of-sync with their consumption patterns. Click here to see the algorithm types available for this plug-in spot.

Budget Plan - True Up Algorithm

On Budget Plan - Main you must define the **True Up Algorithm** used by the Budget True Up background process when it periodically trues up a customer's budget.

If you haven't done so already, you must set up this algorithm in the system. To do this:

- Create a new algorithm (refer to Setting Up Algorithms).
- On this algorithm, reference an Algorithm Type that trues up budget amounts. Click here to see the algorithm types available for this system event.

The system will automatically true up a customer's budget amount every X months (X is defined in **Months for True Up**).

Tender Management

When a payment is received, a tender is created to record what was remitted (e.g., cash, check, credit card). The topics in this section describe control tables that must be set up in order to remit tenders.

> ● **FASTPATH:**
> We strongly recommend Tender Management and Workstation Cashiering before setting up the control tables described in this section.

Setting Up Tender Types

Tender types are used to indicate the method in which the tender was made. A unique **Tender Type** must exist for every type of tender that can be remitted. For example, if you allow cash, checks, direct debits from a checking account, and direct debits from a credit card to be tendered, you'd need the following tender types:

Tender Type	Description	Like Cash	Generate Auto Pay	Require External Source ID	Require Expiration Date	External Type
CASH	Cash	Yes	No	N/A	N/A	N/A
CHEC	Check	No	No	N/A	N/A	N/A
OVUN	Cash drawer - over/under	No	No	N/A	N/A	N/A
DDCH	Direct debit - checking	No	Yes	Yes	No	Checking withdrawal
CRED	Direct debit - credit card	No	Yes	No	Yes	Credit card withdrawal

Go to **Admin** > **Financial** > **Tender Type** to define your tender types.

Description of Page

Enter a unique **Tender Type** and **Description** for the tender type.

Turn on the **Like Cash** switch if this tender type is cash or the equivalent of cash. This indicator controls if the system generates a warning if a cash-only account remits a tender other than cash. It is also used to generate a warning for online cashiers to turn in their tenders when the cash-like amount exceeds the maximum amount balance defined for the tender source.

Turn on **Generate Auto Pay** if this type of tender causes an automatic payment request to be routed to a financial institution. For example, this switch will be on if this tender type is used for direct debits from a customer's checking account (because every tender of this type will have an automatic payment request created when the tender is created).

The following fields are only used for tender types associated with automatic payments:

External Type This field is used by the background process that creates the information that is interfaced to the automatic payment source. Specifically, it controls the record type associated with the different types of automatic payments that are routed to the automated clearinghouse (ACH).

➤ **NOTE:**
The values for this field are customizable using the Lookup table. This field name is EXT_TYPE_FLG.

Require Ext. Src. ID This switch indicates if an Auto-Pay Source that references this type of tender must contain an External Source ID. The External Source ID is the unique identifier of the financial institution to which the automatic payment will be routed.

This switch is typically turned on for tender types associated with checking / saving direct debits. It is turned off for tender types associated with credit card debits (you don't need an external source for a credit card debit, you just need the credit card number).

Expiration Date Required Turn this switch on if an Auto-Pay Option that references an auto-pay source that references this type of tender must also contain an expiration date (e.g., automatic debit / credit cards).

Turn this switch off for tender types associated with checking / saving direct debits.

Tender Authorization Indicates that tenders of a particular type require authorization prior to being created.

Business Object If **Tender Authorization** has a value of Required, a **Business Object** must be specified for the tender type. The primary function of this **Business Object** is to manage the authorization of payment tenders.

●

FASTPATH:
For more information on authorizing credit card payments, refer to the Tender Type - Credit Card with Authorization business object.

Turn on **Allow Cash Back** if the system should automatically calculate a cash back amount when a tender is remitted for this tender type and the amount tender exceeds the amount being paid.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference CI_TENDER_TYPE.

Setting Up Tender Sources

A unique **Tender Source** must exist for every potential source of funds. For example,

- Every cashiering station will have a unique tender source.

➤

NOTE:
If your organization accepts alternate currency payments online, then a tender source must exist for each currency code accepted at the cashiering station.

- Every lock box will have a unique tender source.
- Your remittance processor will have a unique tender source.
- If you allow customers to pay bills automatically (e.g., via EFT), you'll need a tender source for each institution to which you route automatic payment requests. For example, if you route automatic payment requests to the automated clearinghouse (ACH), you'll need a tender source for the ACH.

For example, if you have 3 lock boxes, 2 cash drawers at an area office A, 2 cash drawers at area office B, and a single remittance processor, you'd need the following tender sources:

Tender Source	Type	External Source ID (Lockbox ID)	Default Starting Balance	Currency Code	Suspense Service Agreement
CASH-A01	Cashiering	N/A	150.00	USD	N/A
CASH-A02	Cashiering	N/A	150.00	USD	N/A
CASH-B01	Cashiering	N/A	150.00	USD	N/A
CASH-B02	Cashiering	N/A	150.00	USD	N/A
LB-INDUS	Lockbox	112910-A	N/A	USD	9291019281
LB-COMM	Lockbox	938219-C	N/A	USD	4739837372
LB-RESID	Lockbox	372829-B	N/A	USD	1912910192
REMIT	Lockbox	N/A	N/A	USD	1920038437
ACH	Auto Pay	N/A	N/A	USD	N/A

To set up a tender source, select **Admin** > **Financial** > **Tender Source** > **Add.**

Description of Page

Enter an easily recognizable **Tender Source** and **Description** for the tender source.

Define the **TenderSource Type**. Valid values are: Ad Hoc, Auto Pay, Online Cashiering and Lockbox. The system uses this information to prevent tender controls from different sources from being included under the same deposit control. In other words, you can't mix ad hoc, automatic payment, cashiering and lockbox tenders under the same deposit control.

● **FASTPATH:**
For more information, refer to Maintaining Deposit Controls.

If the source is an external system (e.g., a lockbox or an automatic payment destination), use **External Source ID** to define the unique identifier of the source. The background process that interfaces tenders from this source uses this information to create the appropriate tender control when it interfaces payments from external sources.

If this source is a cash drawer, define the **Default Starting Balance**. This balance is defaulted onto new tender controls and may be overridden.

➤ **NOTE:**
The tender type of the **Start Balance** is defined on the installation record.

If this source is a cash drawer, define the **Max Amount Balance**. When the amount of cash-like tenders in a cash drawer exceeds this balance, a warning is issued to remind the cashier to turn in some of the funds to a tender control.

Define the **Currency Code** of tenders linked to this source. All tenders in a source must be of the same currency.

If this tender source is associated with payments that are interfaced from an external source (e.g., a lockbox or a remittance processor), use Suspense **Service Agreement** to define the service agreement whose account will hold uploaded payments with an invalid account. Refer to Payment Upload Error Segmentation for more information about suspense service agreements. Also note, because the payment upload process simply books payments that reference invalid accounts to the account associated with this service agreement, this account should belong to a customer class with the appropriate payment distribution algorithms. This may entail creating a new customer class that will only be used on these "suspense accounts". This customer class would need the following algorithms:

- We'd recommend using a simple payment distribution algorithm like PYDIST-PPRTY (distribute payment based on SA type's payment priority and the age of the debt).
- We'd recommend using an overpayment distribution algorithm like OVRPY-PPRTY (distribute overpayment to highest priority SA type).

Define the **Bank Code** and **Bank Account** into which the tender source's moneys will be deposited. The bank account defines the distribution code used to build the GL details for the payment. Refer to The Source of GL Accounts on Financial Transactions for more information. Note that the bank code and bank account can later be overwritten when entering Tender Deposits on Deposit Control.

If this tender source is associated with payments that are interfaced from an external source, for example tender sources associated with Auto Pay and Lockbox **Tender Source Types**, the information is also used as follows:

- The payment upload process uses this information to populate the bank and bank account when it creates deposit control records for the tender controls it creates during the interface. Refer to Managing Payments Interfaced From External Sources for more information.
- The automatic payment interface uses this information to populate the bank and bank account when it creates deposit control records for the tender controls it creates during the interface.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference CI_TNDR_SRCE.

➤ **NOTE:**

If your organization accepts alternate currency payments online, then a tender source must exist for each currency code accepted at the cashiering station. When a user adds a tender control the system attempts to default a tender source based on the currency of the deposit control and the tender source(s) define on the user's record.

● **FASTPATH:**

Refer to Alternate Currency Payments for more information.

Automatic Payment Options

If your customers can pay their bills automatically (via direct debit or credit card debits), you'll need to set up the various control tables described in this section.

△ **IMPORTANT:**

Besides the tables described in this section, additional values must also be added to control tables defined under Tender Management. Specifically, refer to Setting Up Tender Types and Setting Up Tender Sources.

● **FASTPATH:**

Refer to Automatic Payments for more information about how automatic payments are handled in the system.

Setting Up Auto-Pay Route Types

Auto Pay Route Types are used to control when and how automatic payment requests are routed to a financial institution, and when the general ledger is impacted. Select **Admin** > **Financial** > **Auto Pay Route Type** to define your route types.

Description of Page

To modify an auto pay route type, simply move to a field and change its value.

To add a new route type, press + to insert a row, then fill in the information for each field. The following fields display:

Route Type The unique identifier of the route type.

Description The description of the route type.

Tender Source The background process that routes automatic payment requests to a financial institution (e.g., the automated clearing house interface) will mark each automatic payment's associated tender with a tender control for audit and control purposes. The following points describe how this happens:

- When the system sees that it's time to send an automatic payment to a financial institution, it looks at the automatic payment's auto-pay source.

- Every auto-pay source references an auto-pay route type.
- Every auto-pay route type references a tender source.
- A **Tender Source** has a tender control for each group of tenders deposited / interfaced together one batch.
- The system marks each automatic payment's associated tender with the latest tender control for the **Tender Source**. The system will create a new tender control each time it routes automatic payments to the tender source. Refer to Managing Payments Interfaced From External Sources for more information about tender source and tender control.

Extract Batch Cd This field defines the background process that interfaces the automatic payment requests to the financial institution.

Autopay Date Calculation Alg This algorithm populates 3 dates associated with the automatic payment: 1) the date the automatic payment will be sent to the financial institution, 2) the date the general ledger will be impacted by the automatic payment, 3) the date of the payment.

If you haven't done so already, you must set up this algorithm in the system. To do this, create a new algorithm (refer to Setting Up Algorithms). On this algorithm, reference an Algorithm Type that populates automatic payment dates. Click here to see the algorithm types available for this plug-in spot.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference CI_APAY_RT_TYPE.

Setting Up Auto-Pay Source Codes

A unique **Auto-Pay Source** must exist for every bank / credit card company / bill payment service that your customer's use as the source of the funds when they sign up for automatic payment. For example,

- Every bank will have a unique auto-pay source.
- Every credit card company will have a unique auto-pay source.

To set up an auto-pay source, select **Admin** > **Financial** > **Auto Pay Source Type** > **Add.**

Description of Page

Enter an easily recognizable **Auto Pay Source Code** and **Description** for the auto-pay source.

The **Source Name** is the name of the financial institution.

When the system creates an automatic payment request, it also creates an associated payment tender. This tender (like all tenders) must have a tender type. This field defines the **Tender Type** associated with this auto-pay source's tenders. Refer to Setting Up Tender Types for more information.

The **External Source ID** is the unique identifier of the financial institution to which the automatic payment will be routed (e.g., the bank routing ID of the bank). This field is typically blank on automatic payments routed to credit card companies because the credit card company doesn't have an external source ID (whereas direct debits from banks must have a bank routing number). Whether this field is required is controlled by the **Tender Type**.

The **Auto Pay Route Type** controls when and how automatic payment requests get routed to a financial institution. It also controls when the general ledger is impacted by the automatic payments financial transaction. Refer to Setting Up Auto-Pay Route Types for more information.

The **Work Calendar** defines the financial institution's workdays. This information is used to determine the date on which automatic payment requests will be sent to the financial institution. Refer to Setting Up External Workday Calendars for more information.

The **Validation Algorithm** defines how the system validates the customer's account ID at the financial institution. If you haven't done so already, you must set up this algorithm in the system. To do this:

- Create a new algorithm (refer to Setting Up Algorithms).

- On this algorithm, reference an Algorithm Type that validates the customer's account ID at the financial institution. Click here to see the algorithm types available for this plug-in spot.

Refer to Account - Auto Pay for more information.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference CI_APAY_SRC.

SEPA Payments

The topics in this section provide background information about the Single Euro Payments Area (SEPA) payment functionality.

> **NOTE: This section is only relevant for some organizations.** The system configuration requirements described in this section are only relevant if your organization is participating in SEPA payment transactions such as direct debit collection.

What Is SEPA?

SEPA (Single Euro Payments Area) is a European Union (EU) integration initiative that is aimed at streamlining processes that are related to cross-border payments. SEPA consists of the EU member states plus a few additional countries. In SEPA, customers can make electronic Euro payments within and across these countries under the same rights and obligations. SEPA payment services are based on global ISO (International Organization for Standardization) standards.

SEPA Direct Debit

The SEPA Direct Debit (SDD) scheme allows a creditor to collect money from the debtor, with prior approval from the debtor. The direct participants are the following:

- Creditor
- Creditor's bank
- Debtor
- Debtor's bank

The debtor and creditor must each hold an account with a payment service provider located within SEPA. The accounts may be in Euro or in any other SEPA currency. However, the transfer of funds between the debtor's bank and the creditor's bank always takes place in the Euro currency. The indirect participants are the following:

- Clearing and Settlement Mechanisms (CSMs) such as an automated clearinghouse
- Intermediary Banks that offer intermediary services to debtor banks and/or creditor banks

SEPA Direct Debit Mandate

The mandate is the consent and authorization that the debtor gives to the creditor, to allow the creditor to initiate direct debit collections. The creditor is responsible for storing the original mandates, together with any amendments relating to the mandate or information regarding its cancellation or lapse. The contents of this section describe how mandates are issued and canceled.

Issuing a Mandate

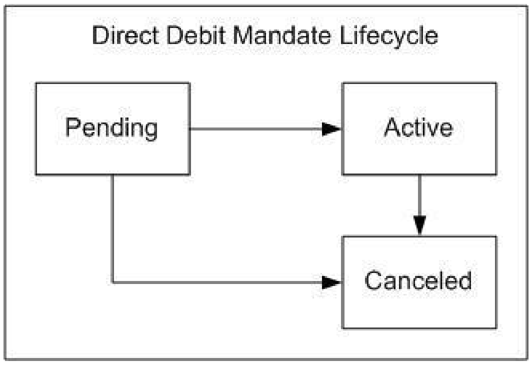
The creditor initiates the issuance of a mandate by sending the mandate form (either paper or electronic) to the debtor with the creditor information filled in. The creditor information includes the creditor's unique identifier as a SEPA creditor. An ISO standard specifies the structure of the creditor identification, which includes country code, a check digit, the creditor's business code and a country-specific identifier for the creditor. The creditor ensures that the mandate form contains the mandatory legal wording and the mandatory set of information, as specified in the standards. The debtor must ensure that the required information is supplied and that the mandate is signed, either in writing or electronically. The mandatory debtor information includes the debtor's international bank account number (IBAN) and bank identifier code (BIC). The creditor assigns a unique reference for the mandate and provides that reference to the debtor before the first initiation of a collection. The debtor can then use both the unique mandate reference and the creditor identification to verify the bank transactions.

Canceling a Mandate

A mandate can be canceled by either the debtor or the creditor, without the involvement of either bank.

The Lifecycle of a Direct Debit Mandate Task

The following diagram shows the possible lifecycle of the Direct Debit Mandate Task business object:



Pending State: The direct debit mandate starts out in a pending state.

Active State: The direct debit mandate is transitioned manually by a user to the active state. Once a mandate is created and activated in the system, the account’s auto pay information must be updated to reflect information on the mandate. This process may be automated within the mandate’s lifecycle and a sample algorithm exists in the demo database to achieve this. This algorithm is also included in the export bundle in the demo database.

Canceled State: The direct debit mandate is transitioned manually by a user to the canceled state.

Configuring the System for SEPA Direct Debits

The following sections describe the setup required if your organization intends to process SEPA direct debit transactions.

Define Auto-Pay Route Type

Configure an auto-pay route type for routing direct debit requests to a financial institution. This auto-pay route type must reference the SEPA Direct Debit Payment Extract (C1-SDDCE) batch process.

Refer to Setting Up Auto-Pay Route Types for more information.

Define Auto-Pay Source Type

Ensure that your auto-pay source codes reference the above auto-pay route type. You must also configure a validation algorithm on the auto-pay source type to ensure the International Bank Account Number (IBAN) defined on the Auto-pay Source of the Account has the correct format.

Refer to Setting Up Auto-Pay Source Codes for more information.

Define a Direct Debit Task Type

Your implementation must configure a **Direct Debit Task Type.** This service task type captures general information that the SEPA extract process needs to create the output file. The base product provides a business object for direct debit task type, **C1-DirectDebitMandateTaskType.**

Note that the direct debit task type is maintained on the Service Task Type portal.

Define SEPA Country Codes

The list of countries and territories that are part of the Single Euro Payment Area (SEPA) must be defined on extendable lookup **C1-SEPACountryCodeLookup**. These country codes are used by the system to validate the creditor identification entered on a direct debit task type.

Note that an export bundle exists in the demo database that includes a sample list of SEPA country codes. This list must be verified by your implementation for accuracy.

Define Mandate Cancel Reasons

A cancel reason is required when canceling a direct debit mandate. Ensure that these status reasons have been configured for the direct debit task business object, **C1-DirectDebitMandateTask.**

Define Navigation Options

The base product includes navigation options for easy access to query and maintain an account’s direct debit mandate. To access these portals, add a new menu item to the account context menu (**CI_CONTEXTACCOUNT**). This menu item should include two menu lines:

- A context menu line for navigation to the direct debit mandate that references a navigation option of **c1DirectDebitMandate.** The system uses the account in context and attempts to find the most recent active mandate for the account. If a mandate exists, navigation is to the service task maintenance portal. If an active mandate does not exist, navigation is to the service task query portal.
- A context menu line for adding a new direct debit mandate that references a navigation option of **c1AddDirectDebitMandate.**

Setting Up Account Auto Pay

The SEPA direct debit extract process retrieves banking details from the account’s auto pay record. Once a mandate is created and activated in the system, the account’s auto pay information must be updated to reflect information on the mandate. This process may be automated within the mandate’s lifecycle and a sample algorithm exists in the demo database to achieve this. This algorithm is also included in the export bundle in demo.

Payment Advices

The topics in this section provide background information about payment advice functionality.

> **NOTE:**
> **This section is only relevant for some organizations.** The system configuration requirements described in this section are only relevant if your organization issues payment advices to the customer instead of initiating electronic funds transfer directly to the customer's bank.

What Is A Payment Advice?

Payment advice is a money order that is established at the initiative of the utility. When a bill is completed, the utility sends the customer a document that indicates a payment amount and the customer's bank details. If the customer agrees to the information on the payment advice, he/she signs it and returns it to the clearinghouse address that is indicated on the payment advice. The clearinghouse, in turn, sends the dated and signed payment advice to the customer's bank, which completes the payment.

Payment Advice vs. Direct Debit

The existing functionality that creates automatic payments is referred to as direct debit processing. Payment advice processing differs from direct debit processing in the way that automatic payments get initiated. With payment advice processing, the usual automatic payment records - i.e. payment event, payment, tender and auto pay clearinghouse staging - are not created. Instead, a payment advice is printed and sent to the customer. The customer sends the approved payment advice directly to the clearinghouse.

> **NOTE:**
> The system does not provide sample processes for extracting and printing payment advice information. Your implementation team would have to create these.

Setting Up The System To Enable Payment Advices

You must set up a Financial Transaction Options Feature Configuration to define parameters that control payment advice functionality.

The following points describe the various **Option Types** that must be defined:

- Payment Advice Functionality Supported. This option controls whether the system allows for payment advice processing.
  - Enter Y if the system should allow for both direct debit and payment advice processing.
  - Enter N if the system should only allow for direct debit processing.

- Default Auto Pay Method. This option is used for defaulting the auto pay method on new account auto pay records.

Refer to Account - Auto Pay for more information on auto pay method.

> **NOTE:**
> The system assumes direct debit processing if the above feature options are not defined.

Credit Card Payments

If your organization accepts credit card payments, you can configure the system to authorize customers' credit card charges in real-time, and perform an authorization reversal (also in real-time) when the credit card payment is canceled. When the authorization web service is not available, you can permit users to enter authorization codes manually so that they can continue processing payments.

Configuring the System for Tender Authorization

The following sections describe the setup required if your organization intends to use the base CyberSource integration tender authorization functionality.

Define the Outbound Message Type

An outbound message type is required for the CyberSource authorization outbound message. This outbound message type must reference the base CyberSource - Credit Card Authorization business object.

An outbound message type is required for the CyberSource reversal outbound message. This outbound message type must reference the base CyberSource - Credit Card Reversal business object.

Define the Message Sender

A Message Sender is required to define how to send messages to CyberSource. Use the context of the Message Sender to define the web service interface.

Define the External System and Configure the Messages

Define an external system and configure the valid outbound message types and the method of communication for each (XAI,Batch or Real Time; Real Time is generally the appropriate choice for credit card authorization). You will also need to select the appropriate XSLs to format both the request and response to the outbound message types for CyberSource.

Define a User

Add a user to hold details required for CyberSource communication. Security information (e.g. Merchant Id, Merchant Reference Code, CyberSource User Name and Password) needed to interface with CyberSource is stored as user characteristics.

Set up the Tender Authorization Algorithm

A Tender Type (BO) - Tender Authorization algorithm must be configured. This algorithm performs a tender authorization or a tender authorization reversal through CyberSource.

Define a Business Object

A business object (BO) must be created for the TENDER TYPE maintenance object. This BO must reference the tender authorization algorithm created.

Define Tender Types

Update the appropriate tender type(s) to denote that authorization is required. The new business object must be specified on the tender type(s).

Tender Authorization - Feature Configuration

If your implementation has a need to prevent users from overriding the automatic tender authorization, then you must set up the Allow Manual Tender Authorization Override option type on the Financial Transaction Options Feature Configuration. The Allow Manual Tender Authorization Override option must have a value of N in order to suppress the Authorization Override checkbox.

FASTPATH:
For more information on credit card payment authorization refer to the Tender Type - Credit Card with Authorization business object.

Non CIS Payments

Payment Templates can be configured for common types of non CIS payment allocations. These templates are used to default the payment distribution and allow non CIS payments to be directly allocated to specific distribution codes.

Setting Up Payment Templates

Payment templates contain the rules that control how non CIS payments are created. You can use a payment template to default the payment distribution for common types of non CIS payments. To set up a payment template, open **Admin** >

Financial > **Payment Template** > **Add.** The topics in this section describe the base-package zones that appear on the Payment Template portal.

Payment Template List

The Payment Template List zone lists every payment template. The following functions are available:

- Click a broadcast button to open other zones that contain more information about the adjacent payment template.
- Click the Add link in the zone's title bar to add a new payment template.

Payment Template

The Payment Template zone contains display-only information about a payment template. This zone appears when a payment template has been broadcast from the Payment Template List zone or if this portal is opened via a drill down from another page. The following functions are available:

- Click the **Edit** button to start a business process that updates the payment template.
- Click the **Delete** button to start a business process that deletes the payment template.
- Click the **Duplicate** button to start a business process that duplicates the payment template.
- Click the **Activate** or **Deactivate** button to start a business process that updates the status of the payment template.

Please see the zone's help text for information about this zone's fields.

Alternate Currency Payments

The topics in this section provide background information about alternate currency payments.

> ➤ **NOTE:**
> **This section is only relevant for some organizations.** The system configuration requirements described in this section are only relevant if your organization accepts payments tendered in a currency other than the customer's currency. If your organization does not accept alternate currency payments, you need only indicate such on the Installation Record; no other setup is required.

What Is An Alternate Currency Payment?

The currency code on the customer's account defines the currency in which the account's financial transactions are expressed. If the customer remits a payment in a different currency, this is referred to as an alternate currency payment. The system enables conversion of the tendered amount to the account's currency and captures the alternate currency and amount, as well as the exchange rate used in the conversion on the payment tender. The payment tender is linked to a tender control that references the alternate currency.

Configuring the System for Alternate Currency Payments

Allowing Alternate Currency Payments

You must set the **Alternate Currency** flag on the Installation Record to Allowed if your organization accepts alternate currency payments. This option controls whether the **Currency Converter** button is displayed when a payment is processed on the payment portal.

Payment Event Business Object

A business object (BO) must be created for the PAY EVENT maintenance object. You must specify this BO as the option value for the CIS Payment Event Add BO option type on the Financial Transaction Options Feature Configuration. This BO must have the Currency Conversion Script BO option defined. This BPA script is invoked when the user clicks on the **Currency Converter** button during CIS payment processing on the payment portal.

➤ **NOTE:**

Currency conversion logic is customizable. The base product includes a script for currency conversion called C1-ConvCurr that's plugged in on the C1-CISPaymentEvent business object. This script converts an alternate currency amount to the account's currency using a bill factor value. The bill factor to use is derived by concatenating the alternate currency code and the account's currency code. For example, if converting US Dollars (USD) to Barbados Dollars (BBD) the bill factor code to use would be USDBBD. Your implementation can change this logic by developing a new script and plugging it into the payment event business object.

Define the User's Tender Sources

Define the tender source(s) for the location (e.g., the specific cash drawer) in which a user's payment tenders are stored during the day. A tender source should be specified for each currency that payments are accepted in. The tender source(s) on the user record are used by the system when a user adds a new tender control. The system attempts to default a tender source on a new tender control based on the deposit control's currency and the tender source(s) defined on the user's record.

Payment Event Distribution

The base-package, by default, creates a single payment for a payment event. If your business requires potentially many payments to be created when payment events are added, you'll need to set up the various control tables described in this section.

● **FASTPATH:**

> Refer to Distributing A Payment Event for more information about how payment event distribution is handled in the system.

Making Payments Using Distribution Rules

As part of this method, one or more distribution details are provided at payment time along with the usual payment and tender information. Each distribution detail record references a distribution rule and a corresponding value. The distribution rule encapsulates the business rules that govern the distribution of the payment amount into payments using the specified value.

The type of value being captured on the distribution detail and the logic that uses it to create payments are defined on the distribution rule.

Rule Value

The primary use of the rule value is to identify the business entity whose balance is to be relieved by creating payment(s). In most cases where the payor account is the same as the payee account it may also used to identify the tender account associated with the payment(s).

Determine Tender Account

The very first step in processing a distribution detail is to identify the tender account (i.e. the payor) associated with the payment. To do that the system calls the Determine Tender Account algorithm defined on the distribution rule providing it with the rule value and other tender information.

Creating Payment(s)

The business logic that distributes a payment amount into one or more payments(s) targeted towards the entity identified by a rule value is held in designated Create Payment algorithms defined on the distribution rule.

Rule Value Can Capture Additional Information

A rule value can also be used to capture additional information provided at payment time, like address information, comments, etc. Obviously payment distribution details with this type of rule value should have a zero payment amount, as they are not real payments. These distribution details end up being linked to a payment event, but unlike other distribution details they do not contribute any payments. You can think of these details as payment event characteristics.

You don't have to set up a Create Payment algorithm for distribution rules intended solely to capture additional payment information.

Setting Up The System To Use Distribution Rules

Setting Up Distribution Rules

Define a Distribution Rule for each payment event distribution method practiced by your business.

Distribution Rule - Main

To set up a distribution rule, navigate to **Admin** > **Financial** > **Distribution Rule** > **Add.**

Description of Page

Enter a unique **Distribution Rule** and **Description** for the distribution rule.

Provide a short and unique **Distribution Rule Label** to be used as rule's name throughout the system.

Characteristic Type defines the type of entity whose balance is relieved by the payment(s) this rule creates. For example, if this rule targets payments(s) towards a specific service agreement, you'd reference a characteristic that its value identifies a service agreement. We use the term "rule value" for the characteristic value.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference CI_DST_RULE

Distribution Rule - Algorithms

Navigate to **Admin** > **Financial** > **Distribution Rule** > **Search** and navigate to the **Algorithms** page to set up the algorithms appropriate for your distribution rule.

Description of Page

The **Algorithms** grid contains algorithms that control important functions. You must define the following for each algorithm:

- Specify the algorithm's **System Event** (see the following table for a description of all possible events).
- Specify the **Algorithm** to be executed when the System Event executes. 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.

The following table describes each **System Event** (note, all system event's are optional and you can define an unlimited number of algorithms for each event).

System Event	Optional / Required	Description
Create Payment	Optional	This algorithm is executed to distribute a payment distribution detail payment amount into one or more payments. Click here to see the algorithm types available for this system event.
Determine Tender Account	Optional	This algorithm is executed to determine the tender account associated with the payment distribution detail. Only one such algorithm may be specified.

Click here to see the algorithm types available for this system event.

Feature Configuration

You must set up a Financial Transaction Options Feature Configuration to define parameters that control various payment event distribution options.

The following points describe the various **Option Types** that must be defined:

- Always Enable Distribution Rule. This option controls whether the system should only use the distribution rule method to add payment events or rather allow both the default method and the distribution rule method to coexist.
  - Enter Y if the system should always use distribution rules. With this setting, navigation to the Payment Event page in add mode opens up the Payment Event Quick Add page (defaulting it to the single payment event dialog). This dialog is designed to create a payment event using distribution rules
  - Enter N if the system should allow both methods. With this setting, navigation to the Payment Event page in add mode opens up the standard Payment Event - Add Dialog that uses the default method to create a payment event. If you want to use the distribution rule method, navigate to the Payment Event Quick Add page from the menu.
- Default Distribution Rule. This option states your default distribution rule that appears throughout the system.

Cancel Reasons

As described in The Financial Big Picture, the various types of financial transactions can be canceled if their financial impact needs to be reversed from the system. Whenever a financial transaction is canceled, a cancel reason must be specified. This section describes the control tables that contain the cancel reason codes.

Setting Up Bill (Segment) Cancellation Reasons

Open **Admin** > **Billing** > **Bill Cancel Reason** to define your bill segment cancellation reason codes.

Description of Page

Enter an easily recognizable **Bill Cancel Reason** and **Description** for the bill cancellation reason.

Only use **System Default** on those reason codes that are placed on bill segments that are automatically canceled by the system. Valid values are: Turn off auto-cancel, Bad estimated read auto-cancel, MDM Corrected Read, and Mass Cancel. The reason code identified as Turn off auto-cancel is placed on bill segments that are automatically canceled when the final bill segment ends before the prior bill (and therefore we have to cancel the prior bill). The reason code identified as Bad estimated read auto-cancel is placed on bill segments that are automatically canceled by the system when it detects that it used an estimated read whose consumption is greater than the next actual read (and therefore we have to cancel the estimated bill segment). The reason code identified as Mass Cancel is placed on bill segments that are canceled as a result of the execution of the Mass Cancellation background process. The reason code identified as MDM Corrected Read is placed on bill segments that are automatically canceled by the system when processing a corrected read notification. Corrected read notifications are received from MDM when reads that were used in bill determinant calculation requests (usage requests) are replaced or modified. This notification may result in the rebill of frozen bill segments. Refer to Usage Requests for more information.

NOTE:

Required values. You must have one reason code defined for each of the System Default values.

Setting Up Payment Cancellation Reasons

Open **Admin** > **Financial** > **Pay Cancel Reason** to define your payment cancellation reason codes.

Description of Page

Enter an easily recognizable **Cancel Reason** and **Description** for the payment cancellation reason.

Turn on the **NSF Charge** switch if the system should invoke the non-sufficient funds (NSF) algorithm when a tender is cancelled using this reason code. Refer to NSF Cancellations for more information.

The next several fields are used to change an account's credit rating or cash-only points if a tender is canceled using the respective reason code.

- Use **Affect Cash-Only Score By** to define how tenders canceled using this reason will affect the account's cash-only score. This should be a positive number. When a customer's cash-only points exceed the cash-only threshold amount defined on the installation record or the account’s CIS Division, the account is flagged as cash only during payment processing and on Control Central.
- Use **Affect Credit Rating By** to define how tenders canceled using this reason will affect the account's credit rating. This should be a negative number. A customer's credit rating is equal to the start credit rating amount defined on the installation record or the account’s CIS Division plus the sum of credit rating demerits that are currently in effect.
- Use **Months Affecting Credit Rating** to define the length of time the demerit remains in effect. This information is used to define the effective period of the credit rating demerit record.

FASTPATH:

For more information, refer to Account - Credit Rating.

NOTE:

The payor gets the credit rating / cash only hit. When you cancel a tender you must specify a cancellation reason. If the cancellation reason indicates a credit rating / cash only demerit should be generated, the system levies the credit rating transaction on the PAYOR's account.

The **System Default Flag** is specified on those cancellation reasons that are placed on payment segments that are automatically cancelled by the system. Valid values are: Re-opened Bill. The Re-opened Bill value is used as follows:

- Payments are automatically created for accounts who pay their bills automatically when their bills are completed.
- If such a bill is reopened before the automatic payment is interfaced to the paying authority, the system automatically cancels the payment. The Re-opened Bill cancellation reason is placed on such payments.

Setting Up Adjustment Cancellation Reasons

Open **Admin** > **Financial** > **Adjustment Cancel Reason** to define your adjustment cancellation reason codes.

Description of Page

Enter an easily recognizable **Cancel Reason** and **Description** for each adjustment cancellation reason.

Miscellaneous Financial Controls

This section describes miscellaneous control tables.

A/P Check Request

Adjustments whose adjustment type is marked with an A/P check request code are interfaced to your A/P system. Your A/P system then cuts the checks.

● **FASTPATH:**
Refer to Controls The Interface To A/P for more information about the accounts payable interface.

You must set up at least one A/P check request code if you want A/P to cut checks.

To set up A/P check request types, open **Admin** > **Financial** > **A/P Request Type.**

Description of Page

Enter an easily recognizable **A/P Request Type** for the accounts payable request type.

Use **Due Days** to define when the check is cut. The cut date is equal to the adjustment date plus due days.

Select a **Payment Method**. Choose from these options:

System Check System check

Where Used

Follow this link to open the data dictionary where you can view the tables that reference CI_APREQ_TYPE.

Billable Charge Line Type

➤ **NOTE:**
Background information. Before using this page, you should be comfortable with the topics described under Setting Up Billable Charge Templates and Uploading Billable Charges.

Billable charge line types will simplify the effort required to interface billable charges from an external system. Each line type contains values that will be defaulted onto the line details associated with the uploaded billable charges. Obviously, this defaulting is possible only if you specify a billable charge line type on the billable charge upload staging lines.

To set up billable charge line types, select **Admin** > **Billing** > **Billable Charge Line Type** > **Add**.

Description of Page

Enter an easily recognizable **Billable Charge Line External Type** and **Description**.

Use **Currency Code** to define the currency to be defaulted onto billable charge upload lines that reference this line type.

Use **Show on Bill** to define the value to be defaulted into the Show on Bill indicator on billable charge upload lines that reference this line type.

Use **App in Summary** to define the value to be defaulted into the App in Summary indicator on billable charge upload lines that reference this line type.

Use **Memo Only, No GL** to define the value to be defaulted into the Memo Only, No GL indicator on billable charge upload lines that reference this line type.

Use **Distribution Code** to define the values to be defaulted into the Distribution Code field on billable charge upload lines that reference this line type.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference CI_BCHG_UP_XTYP.

Payables Cash Accounting

In some areas, taxes and other 3 rd party liabilities are not truly payable until the customer remits payment. We refer to this as "payables cash accounting". This practice should be contrasted with "payables accrual accounting" in which the liability is realized when the bill is created (as opposed to when it is paid).

NOTE:
Value Add Tax (VAT). VAT is a form of taxation common throughout the European Union. It is a common practice to only book the VAT payable when the customer remits payment. This means that most European implementations will use the functionality described in this section.

If your organization does not practice payable cash accounting, you may skip this section as accrual accounting is the system default. If you practice payables cash accounting, the contents of this section describe how to configure the system appropriately.

Accrual versus Cash Accounting Example

The following is an example of the financial events that transpire when a customer is billed and payment is received using accrual accounting.

Event	GL Accounting	Tax Payable Balance
Bill segment created	A/R 110 Revenue <100> Tax Payable <10>	(10)
Payment received	Cash 110 A/R <110>	(10)

In the above example, you'll notice that the payable is booked when the bill is created. Let's contrast this with what takes place if the payable is subject to payables cash accounting.

Event	GL Accounting	Tax Payable Balance	Tax Holding Balance
Bill segment created	A/R 110 Revenue <100> Tax Holding <10>	0	(10)
Payment received	Cash 110 A/R <110> Tax Holding 10 Tax Payable <10>	(10)	0

Notice that when the bill segment is produced, the liability is not booked, rather, the amount of the liability is placed in a "holding" GL account. When the customer pays, the moneys are transferred from the "holding" GL account to the true tax payable account.

NOTE:
Cash accounting is only applicable for liabilities. In the above example, you'll notice that only the tax payable account had cash accounting implications. This is because organizations that practice cash accounting only do it for liability accounts; they never do it for assets, revenue or expenses.

If the above seems simple, consider the following complications that must be considered:

- What happens if a partial payment is received?
- What happens if there are multiple taxes subject to cash accounting rules?
- What happens if the A/R is relieved via a deposit seizure (or transference of a credit balance from another SA)?
- What if, after payment, the original bill segment is cancel/rebilled resulting in a different amount of tax (keep in mind that the payable got booked when the payment was received)?
- What happens if the payment is cancelled?
- What if the payment isn't received and we have to write-off debt?
- What happens if the customer overpays?
- What happens if the customer is allowed to prepay their tax (this is a common practice in the United Kingdom) and then the tax rate changes at billing time?

The above points, and more, are discussed below.

Distribution Code Controls Cash Accounting For A GL Account

NOTE:
If you do not understand the significance of distribution codes, please refer to Setting Up Distribution Codes.

Whether or not cash accounting is used for a specific GL account is defined on HOLDING GL account's distribution code (i.e., the holding GL account references the true payable account).

It is very important that unique payable and holding distribution codes be used for each type of tax subject to cash accounting rules. For example, if you have cash accounting requirements for both value-added tax (VAT) and a climate levy, you would need four distribution codes:

- VAT Payable.
- VAT Holding.
- Climate Levy Payable.
- Climate Levy Holding.

Without unique distribution codes for each payable and holding account, the system cannot keep track of how much of a given tax is being held, awaiting payment.

Bill Segments and Cash Accounting

The contents of this section describe how cash accounting is implemented when bill segments are created.

Bill Segment Financial Transactions Are Not Affected By Cash Accounting

There are NO changes to rate calculation associated with cash accounting. This is because the rate components that calculate tax reference the HOLDING payable distribution codes.

NOTE:
Prepaid taxes - future functionality. If your organization allows customers to prepay taxes in anticipation of a future tax increase (the customers receive the lower rate if they pay in advance), please speak to your account manager for information about when corresponding functionality will be available.

Payment Segments and Cash Accounting

The contents of this section describe how cash accounting is implemented when payment segments are created.

Payment Segment Financial Transaction Algorithms Transfer Holding Amounts to Payable GL Accounts

Logic exists in the pay segment's FT algorithm that transfers amounts from payable holding distribution codes to their respective payable real distribution codes.

FASTPATH:

> Refer to Setting Up Payment Segment Types for how to define the appropriate FT algorithm.

The following table shows what happens to the financial transaction associated with the payment segment for a cash accounting customer.

Event	GL Accounting
Bill segment is created	A/R 110 Revenue <100> Tax Holding <10>
Payment segment relieves receivables	Cash 110 A/R <110>
Additional GL details created when the payment segment FT algorithm transfers the holding amount to a payable account	Tax Holding 10 Tax Payable <10>
Net effect of the above	Cash 110 A/R <110> Tax Holding 10 Tax Payable <10>

How Does The System Know What Amounts To Transfer From Holding To Payables?

When a payment segment is created for an account that is subject to cash accounting processing, the system determines if there is a CREDIT balance for any holding distribution code in respect of the service agreement. If so, it generates additional GL details to transfer moneys from the holding distribution code to the payable distribution code in proportion to the amount of receivables relieved by the payment. Therefore, if 100% of receivables are relieved by the payment segment, 100% of the holding amounts will be transferred to payable distribution codes. Refer to Partial Payments Result In Partial Payables for an example of what happens when a partial payment is created.

Partial Payments Result In Partial Payables

The previous example showed the entire tax holding amount being transferred to the tax payable account. The entire holding amount was transferred because the service agreement was paid in full. If a partial payment is received, only part of the holding amount will be transferred to the payable amount (proportional to the amount of receivables reduced by the payment). An example will help make the point.

Event	GL Accounting	Tax Payable Balance	Tax Holding Balance
Bill segment created	A/R 110 Revenue <100> Tax Holding <10>	0	(10)
Partial payment received	Cash 27.50 A/R <27.50>	(2.50)	(7.50)

	Tax Holding 2.50 Tax Payable <2.50>		

The above example assumes the use of the base product payment segment FT creation algorithm PSEG-AC to transfer the holding amount to the tax payable account. If multiple holding accounts are used, you may want to specify which holding amounts are relieved first. The base product includes an additional payment segment FT creation algorithm C1-FTGL-PSAC that handles booking holding amounts based on a priority.

Partial Payments Using Accounting Priority

To book holding amounts based on a priority, each holding distribution code must be assigned an **Accounting Priority**. When a partial payment is posted, only part of the holding amount will be transferred to the payable amount (proportional to the amount of receivables reduced by the payment). When the holding amount consists of various holding distribution codes with different accounting priorities, the amount to transfer is allocated as follows:

- Holding distribution codes associated with the oldest debt are settled first
- Within the same debt age, holding distribution codes with a higher accounting priority are booked first. If more than one distribution code shares the same priority, the settlement is distributed among them in proportion to the holding account balance

The above logic is handled by the payment segment FT creation algorithm C1-FTGL-PSAC. As an example of how these rules apply, let's assume an implementation practices cash accounting; i.e. revenue, taxes and other third party liabilities are not recognized until payment is received. Also assume the following distribution codes have been configured:

Holding Distribution Code	Description	Cash Accounting Distribution Code	Accounting Priority
HLD-LPC	Late Payment Charge	R-MISC	10
HLD-RGEN	Revenue - Generation Charge	R-GEN	20
HLD-RDIS	Revenue - Distribution Charge	R-DIST	30
HLD-RTRN	Revenue - Transmission Charge	R-TRAN	30
HLD-THRD	3 rd Party Charges	R-THRD	40
HLD-VAT	VAT	A/P-VAT	90

Assume a customer has an outstanding third party charge with an arrears date of 2/Jan/2009:

Event	GL Accounting	SA's Payoff Balance	Holding Balances					
			HLD-LPC	HLD-RGEN	HLD-RDIS	HLD-RTRN	HLD-THRD	HLD-VAT
Unpaid Amount	A/R 50 HLD-HRD <45> HLD-VAT <5>	50	0	0	0	0	(45)	(5)

A bill is created for a customer and the result of the bill's financial transactions (an LPC adjustment in the amount of 10 and a bill segment in the amount of 127) include the following FT GL lines (both financial transactions have an arrears date of 15/Jan/2009):

Event	GL Accounting	SA's Payoff Balance	Holding Balances

			HLD-LPC	HLD-RGEN	HLD-RDIS	HLD-RTRN	HLD-THRD	HLD-VAT
Bill segment created	A/R 127 HLD-RGEN <15> HLD-RDIS <20> HLD-RTRN <55> HLD-THRD <10> HLD-VAT <27>	177	0	(15)	(20)	(55)	(55)	(32)
Adjustment created	A/R 10 HLD-LPC <10>	187	(10)	(15)	(20)	(55)	(55)	(32)

No payment is received prior to the next bill. The result of the next bill's financial transaction (a bill segment in the amount of 100) includes the following FT GL lines (this financial transaction has an arrears date of 16/Feb/2009):

Event	GL Accounting	SA's Payoff Balance	Holding Balances					
			HLD-LPC	HLD-RGEN	HLD-RDIS	HLD-RTRN	HLD-THRD	HLD-VAT
Bill segment created	A/R 100 HLD-RGEN <15> HLD- RDIS <20> HLD- RTRN <45> HLD- THRD <10> HLD-VAT <10>	287	(10)	(30)	(40)	(100)	(65)	(42)

The following shows the result if a customer makes a payment on 20/Feb/2009. At payment time we'll build a table of holding amounts by accounting priority and debt age as follows:

Distribution Code	HLD-LPC	HLD-RGEN	HLD-RDIS	HLD- RTRN	HLD-THRD	HLD- VAT
& Priority	(10)	(20)	(30)	(30)	(40)	(90)
Debt Age						
4 days old		(15)	(20)	(45)	(10)	(10)
36 days old	(10)	(15)	(20)	(55)	(10)	(27)
49 days old					(45)	(5)

Examples of Partial Payments Using Accounting Priority

The examples below assume a customer has the financial history described above and attempts to illustrate the financial effect when a payment is made.

- Example 1 - Customer Pays In Full
- Example 2 - Customer Makes a Partial Payment
- Example 3 - Customer Makes a Partial Payment

Example 1 - Customer Pays In Full

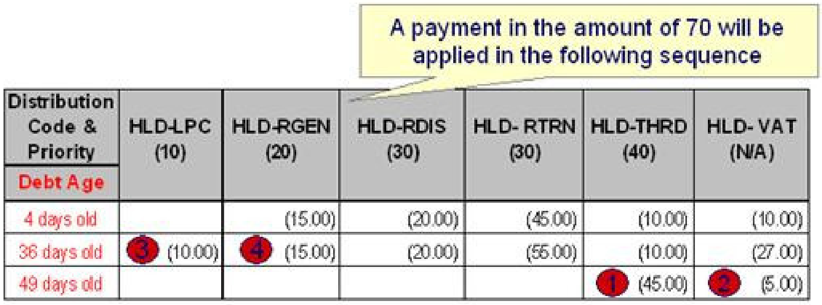
Assume the customer makes a payment in the amount of 287. This amount is sufficient to satisfy all holding amounts, so the payment will have the following financial effect:

Event	GL Accounting	SA's Payoff Balance	Holding Balances					
			HLD-LPC	HLD-RGEN	HLD-RDIS	HLD-RTRN	HLD-THRD	HLD-VAT
Payment received	Cash 287 A/R <287> HLD-LPC 10 R-MISC <10> HLD-RGEN 30 R-GEN <30> HLD- RDIS 40 R-DIST <40> HLD- RTRN 100 R-TRAN <100> HLD- THRD 65 R-THRD <65> HLD-VAT 42 A/P-VAT <42>	0	0	0	0	0	0	0

Top of the Page

Example 2 - Customer Makes a Partial Payment

Assume the same financial history described above for a customer and a partial payment in the amount of 70 is made. This amount is not sufficient to satisfy the total holding amounts of 287, so the system will start settling held amounts starting with distribution codes with the oldest debt first from highest priority until the payment amount is exhausted.



A payment in the amount of 70 will be applied in the following sequence

Distribution Code & Priority Debt Age	HLD-LPC (10)	HLD-RGEN (20)	HLD-RDIS (30)	HLD- RTRN (30)	HLD-THRD (40)	HLD- VAT (N/A)
4 days old		(15.00)	(20.00)	(45.00)	(10.00)	(10.00)
36 days old	❸ (10.00)	❹ (15.00)	(20.00)	(55.00)	(10.00)	(27.00)
49 days old					❶ (45.00)	❷ (5.00)

The following describes how the holding amounts will be booked as a result of this partial payment:

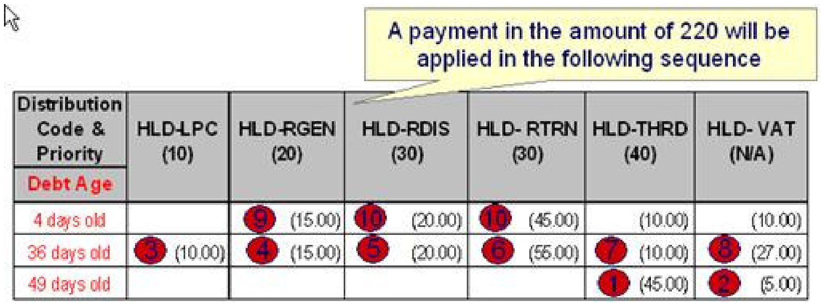
- Settle oldest debt first (49 days old), i.e. 3rd Party Charges (HLD-THRD) and VAT (HLD-VAT). Note that even though these holding accounts have the lower accounting priorities, they are booked first because they have the oldest debt. An amount of 20 now remains on the partial payment.
- Next, we'll settle the 36 days old debt from the highest priority:
  - Late Payment Charge (HLD-LPC) in the amount of 10. An amount of 10 now remains on the partial payment
  - Revenue - Generation Charge (HLD-RGEN) gets the remaining payment amount of 10
- So, this partial payment in the amount of 70 will result in the following financial effect:

Event	GL Accounting	SA Balance	Holding Balances					
			HLD-LPC	HLD-RGEN	HLD-RDIS	HLD-RTRN	HLD-THRD	HLD-VAT
Payment received	Cash 70 A/R <70> HLD-LPC 10 R-MISC <10> HLD-RGEN 10 R-GEN <10> HLD- THRD 45 R-THRD <45> HLD-VAT 5 A/P-VAT <5>	217	0	(20)	(40)	(100)	(20)	(37)

Top of the Page

Example 3 - Customer Makes a Partial Payment

Assume the same financial history described above for a customer and a partial payment in the amount of 220 is made. This amount is not sufficient to satisfy the total holding amounts of 287, so the system will start settling held amounts starting with distribution codes with the oldest debt first from highest priority until the payment amount is exhausted.



A payment in the amount of 220 will be applied in the following sequence

Distribution Code & Priority Debt Age	HLD-LPC (10)	HLD-RGEN (20)	HLD-RDIS (30)	HLD- RTRN (30)	HLD-THRD (40)	HLD- VAT (N/A)
4 days old		9 (15.00)	10 (20.00)	10 (45.00)	(10.00)	(10.00)
36 days old	3 (10.00)	4 (15.00)	5 (20.00)	6 (55.00)	7 (10.00)	8 (27.00)
49 days old					1 (45.00)	2 (5.00)

The following describes how the holding amounts will be booked as a result of this partial payment:

- Settle oldest debt first (49 days old), i.e. 3rd Party Charges (HLD-THRD) and VAT (HLD-VAT). Note that even though these holding accounts have the lower accounting priorities they are booked first because they have the oldest debt. An amount of 170 now remains on the partial payment.
- Next, we'll settle the 36 days old debt from the highest priority, i.e. Late Payment Charge (HLD-LPC), Revenue - Generation Charge (HLD-RGEN), Revenue - Distribution Charge (HLD-RDIS), Revenue - Transmission Charge (HLD-RTRN), 3rd Party Charges (HLD-THRD) and VAT (HLD-VAT). An amount of 33 now remains on the partial payment.
- Next we'll settle the 4 day old debt from the highest priority:
  - Revenue - Generation Charge (HLD-RGEN) in the amount of 15. An amount of 18 now remains on the partial payment
  - The two holding accounts at the next priority have an outstanding amount of 65. Since the remainder of the payment is not enough to satisfy this amount, the remainder of the payment is prorated amongst HLD-RDIS and HLD-RTRN as follows:
    - (Remaining Pay Amount / Total Outstanding Holding Amount) * Holding Account Amount
    - So for the Revenue - Distribution Charge (HLD-RDIS) holding account the amount booked will be (18/65 * 20) = 5.54
- So, this partial payment in the amount of 220 will result in the following financial effect:

Event	GL Accounting	SA Balance	Holding Balances					
			HLD-LPC	HLD-RGEN	HLD-RDIS	HLD-RTRN	HLD-THRD	HLD-VAT
Payment received	Cash 220 A/R <220> HLD-LPC 10 R-MISC <10> HLD-RGEN 30 R-GEN <30> HLD-RDIS 25.54	67	0	0	(14.46)	(32.54)	(10)	(10)

R-DIST
<25.54>

HLD-
RTRN
67.46

R-TRAN
<67.46>

HLD-
THRD
55

R-THRD
<55>

HLD-
VAT32

A/P-VAT
<32>

Adjustments That Behave Like Payments

There are several types of adjustments that behave just like payments (in respect of payables cash accounting). Consider the following events:

- Seizing a deposit (i.e., transferring a credit from a deposit service agreement to a regular service agreement)
- Overpayments transferred from one service agreement to another

The above events should cause the system to transfer holding amounts to true payable amounts (notice that the above examples are all transfer adjustments).

However, there are many other adjustments that should NOT behave like payments. You control how the adjustment works by selecting the appropriate FT algorithm when you set up adjustment types (refer to ADJT-AC and ADJT-TC for a description of the base package algorithms that cause the holding amounts to be manipulated in proportion to the amount of receivable being adjusted; and to C1-FTGL-ADAC and C1-FTGL-ADTC for the base package algorithms that take **Accounting Priority** into consideration). In other words, there are adjustment FT algorithms that cause the transference of holding payable amounts to real payable amounts when the A/R balance is decreased by the adjustment.

> **NOTE:**
> **Cash refunds can behave like "anti-payments".** In addition to the above examples of transfer adjustments behaving like payments, you should be aware that cash refunds may impact your holding and true payable balances. Refer to Cash Refunds for more information.

Overpayment Of Taxes Due To Cancel/Rebills

Let’s assume a cancel / rebill occurs after a payment is received and the net effect of the cancel / rebill is that the customer has overpaid their taxes.

Event	GL Accounting	Tax Payable Balance	Tax Holding Balance

Bill segment created	A/R 110 Revenue <100> Tax Holding <10>	0	(10)
Payment received	Cash 110 A/R <110> Tax Holding 10 Tax Payable <10>	(10)	0
Cancel	A/R <110> Revenue 100 Tax Holding 10	(10)	10
Rebill	A/R 27.50 Revenue <25> Tax Holding <2.50>	(10)	7.50

You'll notice that the amount payable to the taxing authority still indicates $10 (the amount of tax that was paid by the customer). However, you'll notice that the tax holding balance is 7.50 (debit). This looks a bit odd, but it's correct. Remember that at this point, the customer has a credit balance of $75 and this will be whittled down as successive bills are produced as shown below. Note: refer to Cash Refunds for an example of what happens if you refund the credit with a check rather than letting it whittle down.

Event	GL Accounting	Tax Payable Balance	Tax Holding Balance
		(10)	7.50
Bill segment created	A/R 55 Revenue <50> Tax Holding <5>	(10)	2.50
Bill segment created	A/R 110 Revenue <100> Tax Holding <10>	(10)	(7.50)

In the unlikely event of a payment being received while the tax holding has a debit balance, nothing will be done in respect of transferring funds from holding to payable (there is nothing to transfer).

Cash Refunds

If you refund moneys to a cash accounting customer, it's important to do the opposite of what was done when the payment was received (i.e., you need to transfer the payable back to the holding account). The following example should help clarify this situation (this example shows a refund due to a credit balance that occurred as a result of a cancel/rebill).

Event	GL Accounting	Tax Payable Balance	Tax Holding Balance	SA's Payoff Balance
Bill segment created	A/R 110 Revenue <100> Tax Holding <10>	0	(10)	110
Payment received	Cash 110	(10)	0	0

	A/R <110> Tax Holding 10 Tax Payable <10>			
Cancel	A/R <110> Revenue 100 Tax Holding 10	(10)	10	(110)
Rebill	A/R 27.50 Revenue <25> Tax Holding <2.50>	(10)	7.50	(82.50)
Payment refunded (via an A/P adjustment)	Cash <82.50> A/R 82.50 Tax Holding <7.50> Tax Payable 7.50	(2.50)	0	0

We understand this is tricky, but consider this - when a cash accounting customer makes a payment, the system transfers tax holding CREDIT balances to tax payable distribution codes in proportion to the amount of the receivable DEBIT amount that was reduced by the payment. Therefore, when cash is returned to the customer, the system should transfer tax holding DEBIT balances to tax payable distribution codes in proportion to the amount of the receivable CREDIT that was reduced by the refund.

➤ **NOTE:**
The above takes place when an A/P adjustment is created if the related adjustment type references the appropriate FT algorithm (refer to ADJT-AC and **C1-FTGL-ADAC** for a description of the adjustment FT algorithms used for adjustments that behave like payments).

Overpayments

An overpayment, by definition, does not "match" to open items. However, the match type algorithms supplied with the base package will result in a balanced match event if an overpayment is made. The following points explain how this is achieved:

- The base package's match type algorithms will distribute the payment until the customer's current debt is satisfied.
- The amount of the overpayment will be kept on a separate SA (this only happens if you plug-in the appropriate Overpayment Distribution algorithm on your customer classes). Refer to Overpayment Segmentation for more information.
- When the payment is frozen, the payment segments that satisfy current debt will be matched against their respective open-items. The payment segment used to book the overpayment (on the overpayment SA) will not be matched.
- When future bills are completed, the credit balance on this "overpayment SA" will be transferred to the "real SAs" when future bills are completed (if you have plugged in the appropriate bill completion algorithm on the overpayment SA's SA type). If the overpayment satisfies all newly calculated charges, a match event is created that matches the new charges against the funds transferred from the overpayment SA. Refer to When Are Match Events Created for information about how the system creates match events at bill completion time when the new charges on the bill are satisfied by other credits (overpayments, deposit refunds, etc.).
- At some point in the future, the overpayment will be exhausted (i.e., all funds will be transferred to "real SAs"). At that point in time, the overpayment SA will close (assuming you set up the overpayment SA's SA type as a "one time"). At close time, the system creates a match event that matches the original overpayment payment segment with the

adjustments that were used to transfer funds to the "real SAs". Refer to When Are Match Events Created for information about how the system creates match events when a SA closes.

Write Down Adjustment

Writing down debt is very different from writing off debt. When you write down debt, you are removing the receivable with no expectation of it being paid. For example, most organizations write down small debit and credit balances as part of their write-off process (e.g., they don't send a very small amount to a collection agency).

Let's run through an example to illustrate this:

Event	GL Accounting	Tax Payable Balance	Tax Holding Balance	SA's Payoff Balance
Bill segment created	A/R 110 Revenue <100> Tax Holding <10>	0	(10)	110
Payment received	Cash 109.50 A/R <109.50> Tax Holding 9.95 Tax Payable <9.95>	(9.95)	.05	0.50
Write down cash accounting debt	Tax Holding 0.05 Write Down Expense 0.45 A/R <0.50>	(9.95)	0	0

In order to achieve the above, you must set up an adjustment type that references a special financial transaction algorithm (refer to non-accrual accounting write down algorithms ADJT-AD and **C1-FTGL-AD** for more information). This algorithm will reduce / increase the receivable balance accordingly AND cause any holding amounts to be set to zero. This adjustment type should be referenced on your write-down algorithm that is referenced on your write-off controls.

Write-Offs

At write-off time we may refund credit balances. The refunding of credit balances is handled by A/P adjustments and these have cash accounting processing as described under Cash Refunds.

If we have to write-off debt, holding balances are relieved in proportion to the amount of debt that is written off (as usual). It's important to understand that for this to work, you must set up the system as follows:

- The tax holding distribution codes must have their override distribution switch turned on.
- The distribution code on the SA type associated with the service agreement to which the written-off payables are transferred must be the REAL payable distribution codes. This is important so that if the customer pays after the payables are reversed, we will be able to debit cash and credit the REAL payable distribution code.

Let's run through an example to illustrate this.

Event	Normal SA GL Accounting	Write Off Revenue SA GL Accounting	Reverse Liabilities SA GL Accounting
Bill segment created	A/R 110 Revenue <100>		

	Tax Holding <10>		
Write Off Time			
Reverse the held payables	Xfer 10 A/R <10>		Tax Holding 10 Xfer <10> Note, the tax holding only gets debited if you have turned on the override at write-off switch on its distribution code
Write off revenue	Xfer 100 A/R <100>	Write Off Expense 100 Xfer <100>	
If the customer subsequently pays		Cash 100 Write Off Exp <100>	Cash 10 Tax Payable <10> Note, the tax payable only gets credited if the SA type's distribution code has been defined as such

Deferred Accrual Accounting

Some implementations use a hybrid accounting method that combines cash and accrual accounting. In this case revenue, taxes, etc. are recognized on the earlier of the bill due date or the date payment is received. In this scenario, the cash accounting method is used up until the bill's due date, at which time the accrual method is enforced (let's call this "deferred accrual accounting"). A simpler flavor of deferred accrual accounting is when the revenue and liability recognition is done solely at bill due date regardless of when the payment is made.

Deferred accrual accounting affects distribution codes identified as **Use For Non-Accrual Accounting** with an associated **Accounting Method** of either Payable on Earlier of Payment or Due Date or Payable on Due Date. The system accomplishes the holding amount settlement on the bill due date using a customer class post bill completion algorithm C1-CR-BLRVWS that creates a bill review record to be processed on the bill's due date. The bill review batch process then analyzes these bill review records on the bill due date. If a bill review record is due for processing, the algorithm checks the outstanding balance of the holding accounts on each SA linked to the bill and creates a settlement adjustment for each SA.

Deferred Accrual Accounting Examples

The examples below illustrate the financial transactions that transpire under these different scenarios.

Example 1 - Payable On Due Date, Customer Pays In Full

The following is an example of the financial events that transpire when a customer is billed and full payment is received prior to the bill due date. The accounting method in this case is Payable On Due Date.

Event	GL Accounting	Tax Payable Balance	Tax Holding Balance
Bill segment created	A/R 110	0	(10)

	Revenue <100> Tax Holding <10>		
Payment received	Cash 110 A/R <110>	0	(10)
Adjustment created on bill due date	Tax Holding 10 Tax Payable <10>	(10)	0

Example 2 - Payable On Due Date, Customer Does Not Pay

In the following example a customer is billed and no payment is received prior to the bill due date. The bill review batch process is responsible for creating the settlement adjustment for any outstanding holding amounts on the bill's due date. The accounting method in this case is Payable On Due Date.

Event	GL Accounting	Tax Payable Balance	Tax Holding Balance
Bill segment created	A/R 110 Revenue <100> Tax Holding <10>	0	(10)
Adjustment created on bill due date	Tax Holding 10 Tax Payable <10>	(10)	0

Example 3 - Payable On Due Date, Customer Makes A Partial Payment

In the following example a customer is billed and a partial payment is received prior to the bill due date. The bill review batch process is responsible for creating the settlement adjustment for any outstanding holding amounts on the bill's due date. The accounting method in this case is Payable On Due Date.

Event	GL Accounting	Tax Payable Balance	Tax Holding Balance
Bill segment created	A/R 110 Revenue <100> Tax Holding <10>	0	(10)
Payment received	Cash 27.50 A/R <27.50>	0	(10)
Adjustment created on bill due date	Tax Holding 10 Tax Payable <10>	(10)	0

Example 4 - Payable On Earlier Of Payment Or Due Date, Customer Pays In Full

In the following example a customer is billed and full payment is received prior to the bill due date. The accounting method in this case is Payable On Earlier Of Payment Or Due Date.

Event	GL Accounting	Tax Payable Balance	Tax Holding Balance
Bill segment created	A/R 110 Revenue <100> Tax Holding <10>	0	(10)
Payment received	Cash 110 A/R <110> Tax Holding 10 Tax Payable <10>	(10)	0

Example 5 - Payable On Earlier Of Payment Or Due Date, Customer Does Not Pay

In the following example a customer is billed and no payment is received prior to the bill due date. The bill review batch process is responsible for creating the settlement adjustment for any outstanding holding amounts on the bill's due date. The accounting method in this case is Payable On Earlier Of Payment Or Due Date. Note that if a payment is subsequently received after the settlement adjustment has been created, it's financial transaction(s) will not have any impact on the holding or liability accounts as these have already been booked.

Event	GL Accounting	Tax Payable Balance	Tax Holding Balance
Bill segment created	A/R 110 Revenue <100> Tax Holding <10>	0	(10)
Adjustment created on bill due date	Tax Holding 10 Tax Payable <10>	(10)	0

Example 6 - Payable On Earlier Of Payment Or Due Date, Customer Partially Pays

In the following example a customer is billed and a partial payment is received prior to the bill due date. The bill review batch process is responsible for creating the settlement adjustment for any outstanding holding amounts on the bill's due date. The accounting method in this case is Payable On Earlier Of Payment Or Due Date. Note that if a payment is subsequently received after the settlement adjustment has been created, it's financial transaction(s) will not have any impact on the holding or liability accounts as these have already been booked.

Event	GL Accounting	Tax Payable Balance	Tax Holding Balance
Bill segment created	A/R 110 Revenue <100> Tax Holding <10>	0	(10)
Payment received	Cash 27.50	(2.50)	(7.50)

	A/R <27.50> Tax Holding 2.50 Tax Payable <2.50>		
Adjustment created on bill due date	Tax Holding 7.50 Tax Payable <7.50>	(10)	0

Payment Cancellations and Deferred Accrual Accounting

If a payment was responsible for transferring moneys from the holding distribution code to the payable distribution code, it stands to reason that if the payment is cancelled, it results in the reversal of this transfer from the holding distribution code to payable distribution code. If deferred accrual accounting is used and the payment is cancelled after the bill due date, the holding amounts that were transferred should remain booked.

Assume, for example, that the payment below was cancelled after the bill due date:

Event	GL Accounting
Payment received	Cash 110 A/R <110> Tax Holding 10 Tax Payable <10>

At cancellation, the above entry will be reversed, reinstating the balance in the holding account:

Event	GL Accounting
Payment cancelled after bill due date	Cash <110> A/R 110 Tax Holding <10> Tax Payable 10

However, since the bill's due date has passed, the holding account needs to be booked. For open item accounts, the system comes with a customer class FT freeze algorithm (C1-PR-CA-RVS) that creates a bill review schedule for the affected bill, if one does not already exist. When the bill review batch process next runs, it checks the outstanding balance of the holding accounts on each SA linked to the bill and creates a settlement adjustment for each SA.

Event	GL Accounting
Adjustment created	Tax Holding 10 Tax Payable <10>

Note that this solution is only applicable to open item accounting where the bill matched to the payment can be determined. If balance forward accounting is practiced, the bill or bills that the payment applied to cannot be determined. In this case, the next bill review record created for the account as part of billing will cause the balances of the holding accounts to be analyzed and the settlement will catch up at that point.

Open Item Accounting

The topics in this section provide background information about open-item accounting.

➤

NOTE:

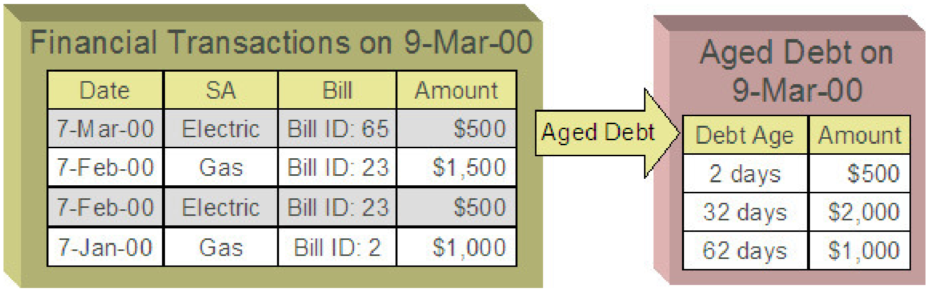
This section is only relevant for some organizations. The system configuration requirements described in this section are only relevant if your organization practices open-item accounting. If your organization practices balance-forward accounting, you need only indicate such on your customer classes; no other setup is required. Refer to Open Item Versus Balance Forward Accounting for more information about these two accounting practices.

Open-Item Versus Balance-Forward Accounting

If you practice open-item accounting, you match payments against bills. The term "open-item accounting" is used to describe this accounting practice because:

- Payments are matched against "open items" (i.e., unpaid bills and adjustments)
- Only unmatched bills and adjustments (i.e., open items) affect aged debt.

Contrast open-item accounting with "balance-forward" accounting - in a balance-forward world, payments are not matched to bills. Rather, payments implicitly relieve a customer's oldest debt. For example, consider the following unpaid financial transactions that exist for an account and the resultant aged debt.



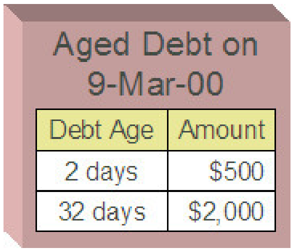
Financial Transactions on 9-Mar-00

Date	SA	Bill	Amount
7-Mar-00	Electric	Bill ID: 65	$500
7-Feb-00	Gas	Bill ID: 23	$1,500
7-Feb-00	Electric	Bill ID: 23	$500
7-Jan-00	Gas	Bill ID: 2	$1,000

Aged Debt on 9-Mar-00

Debt Age	Amount
2 days	$500
32 days	$2,000
62 days	$1,000

In a balance-forward world, if a $1,000 payment was made on 9-Mar-00, the customer's aged debt would look as follows:

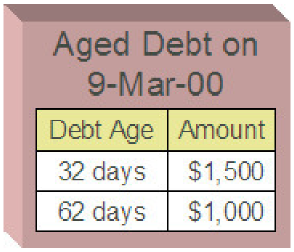


Aged Debt on 9-Mar-00

Debt Age	Amount
2 days	$500
32 days	$2,000

Notice how the $1,000 payment relieves the 62 day old debt - it does this because, in a balance-forward world, payments payoff oldest debt first.

However, let's assume the customer wants the payment to settle his electric debt (e.g., because he disagrees with the gas bills). If you could match the $1,000 payment to the two electric bills (i.e., open-item accounting exists), the customer's aged debt would look as follows:



Aged Debt on 9-Mar-00

Debt Age	Amount
32 days	$1,500
62 days	$1,000

In sum,

- In an open-item world, payments are matched to bills and only unpaid bills and adjustments (i.e., open items) affect aged debt.
- In a balance-forward world, payments are not matched to bills and therefore a customer's aged debt is computed by aging debits (bills and adjustments) and then relieving the oldest debits using credits (payments and adjustments).

NOTE:
Financial Transactions and Bills. In an open-item world, only bill segments and adjustments are presented on a bill. When a bill is completed, only those bill segments and adjustments to be presented are swept onto a bill. Payment and payment cancellation FTs, bill segment FTs canceled before bill completion together with their corresponding bill segment cancellation FTs, and adjustment FTs marked as do not show on bill are not swept onto a bill. An adjustment's adjustment type and its algorithms determine if its FT will show on a bill by default.

Accounting Method Defined On Your Customer Classes

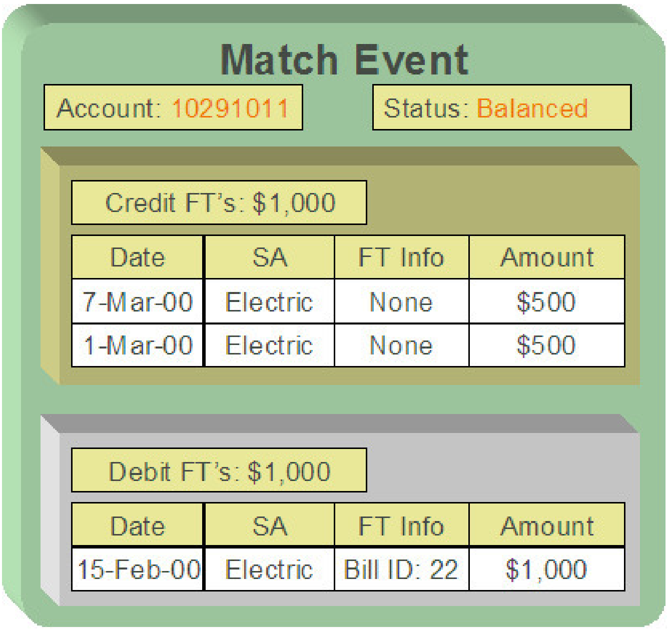
You define the type of accounting method that is practiced (balance-forward versus open-item) on your customer classes. For example, residential customers can practice balance-forward accounting whereas industrial / commercial customers can practice open-item accounting.

Match Events

Match events are used to match open-items (i.e., debit and credit financial transactions) together. The topics in this section provide an overview of match events.

Match Events Match Debit FTs To Credit FTs

For open-item customers, the system matches credit financial transactions (FT's) to debit FT's under a match event. The following is an example of a match event associated with two $500 payments that satisfy the debt associated with one bill (on February 2000).



Match Event

Account: 10291011

Status: Balanced

Credit FT's: $1,000

Date	SA	FT Info	Amount
7-Mar-00	Electric	None	$500
1-Mar-00	Electric	None	$500

Debit FT's: $1,000

Date	SA	FT Info	Amount
15-Feb-00	Electric	Bill ID: 22	$1,000

Notice the following:

- The match event matches 2 credit FT's against a single debit FT. A match event may contain an unlimited number of FT's.
- The match event contains FT's associated with a single account. While the FT's under a match event may belong to multiple service agreements, all FT's under a match event must belong to the same account.
- The match event only contains bill segments that belong to a single bill. If you mix multiple bills under a single match event, then an individual bill balance cannot be properly determined when partial payments exist.
- The status of the match event is balanced. This is because the sum of the debits equals the sum of the credits. If debits do not equal credits, the status of the match event would be open and the various FT's would still affect the customer's aged debt. Refer to Match Event Lifecycle for more information.

WARNING:
It is strongly encouraged that you refrain from mixing multiple bills on a single match event. If you stick by the rule of "just one bill per match event" you will then be able to determine the outstanding balance of a partially paid bill (see the bill page, bill summary section). However, if you mix more than one bill under a match event, a particular bill's balance may become indeterminate. Algorithm types have been provided which help to enforce this rule of "one bill per match event", please refer to Match By Bill, Pay Oldest Bill First for an example of a matching algorithm that enforces this notion.

When Are Match Events Created?

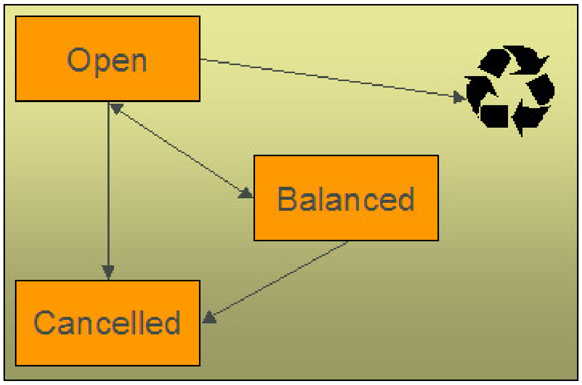
The following points describe when match events are created for open-item accounts:

NOTE:
Match events are only created for open-item accounts (i.e., those accounts with a customer class that indicates open-item accounting is practiced). Match events may not be created for balance-forward accounts.

- The system can create one or many match events when a payment is added. This match event matches the payment's credit FT's with the debit and credit FT's from bill segments and adjustments. The FT's that are linked to the match event are controlled by the payment's **match type** and **match value** (payments made by open-item customers must reference a match type and match value). Refer to Payments And Match Events for more information.
- The system may create a match event when any type of financial transaction is cancelled. This match event groups together the original FT with its cancellation FT. Refer to How Are Match Events Cancelled? for more information.
- The system creates a match event when a bill is completed for customers that pay automatically (i.e., direct debit customers). The match event groups together the bill's new charges against the automatic payment's payment segments.
- The system creates a match event when a bill is completed where the new charges are offset by other financial transactions. For example,
    - Consider a bill that contains a deposit refund. If the sum of the deposit refund equals or exceeds the amount of the bill, the bill's FT's can be matched against the debit refunds FT's. Refer to Refunding Deposits for more information about deposit refunds.
    - Consider a bill whose new charges are offset by a previous overpayment. Refer to Over Payments for more information about overpayments.
    - **FASTPATH:**
      Refer to Bill Lifecycle for more information about what happens during bill completion.
- The system creates a match event when a service agreement closes and the service agreement has unmatched FT's. For example,
    - Consider a deposit service agreement that closes when the deposit is refunded to the customer. The system will create a match event with the deposit SA's FT's (the original credit and the debits used to refund the deposit) when the deposit SA closes (i.e., when its credit balance falls to zero). Refer to Refunding Deposits for more information about deposit refunds.
    - Consider a service agreement for utility debt that is written off. This service agreement closes when the system creates transfer adjustments to transfer the utility debt to a write-off service agreement (or writes down the debt). The system creates a match event to match the original debt to the transfer adjustments used to write-off the debt. Refer to How Is Debt Financially Written Off for more information about write-off processing.
- A user can create a match event manually at any time. Manual match events would be created under a variety of situations. For example:
    - If a customer disputes a charge. Refer to Disputing Items for more information about disputes.
    - To handle unusual situations when the system is unable to automatically match FT's together.

Match Event Lifecycle

The following diagram shows the possible lifecycle of a match event:



Match events are initially created in the open state. Financial transactions (FT's) linked to open match events affect arrears, but not in an open-item fashion. Rather, FT's linked to open match events affect arrears in a balance-forward fashion. Refer to Open Item Versus Balance Forward Accounting for more information about these two accounting methods.

A user may delete an open match event. When an open match event is deleted, its FT's may be linked to other match events.

The system automatically changes an open event's status to balanced when the sum of the debit financial transactions (FT's) equals the sum of the credit FT's for each SA on the match event. It's worth stressing that a match event may contain FT's from many SAs and each SA's FT's must sum to zero before the match event can become balanced.

A user may re open a balanced event (by adding / removing FT's so that the match event becomes unbalanced).

A user may cancel a balanced or open match event. Refer to How Are Match Events Cancelled? for more information about cancellation.

Payments And Match Events

As described under When Are Match Events Created?, the system creates a match event when a payment is added for an open-item account. The system uses the payment's **match type** and **match value** to determine the FT's (e.g., bill segments and adjustments) that will be matched with the payment's FT's (i.e., the payment segments).

Another way to think of this is as follows:

- When most payments are distributed, the system calls the payment distribution algorithm that is plugged-in on the account's customer class.
- However, a payment that is made in respect of a specific bill requires a different distribution algorithm because the payment should only be distributed amongst the debt associated with the specific bill being paid. This is accomplished by referencing a match type / match value on the payment. The match type references the appropriate payment distribution algorithm. This algorithm is used rather than the customer class distribution algorithm.

For example, if a payment were made in respect of bill ID 192910192101, this payment would reference a match type of bill ID and a match value of 192910192101. At payment distribution time, the system calls the override payment distribution algorithm associated with this match type. The base package bill ID distribution algorithm does several things:

- It distributes the payment amongst service agreements associated with the bill.
- It creates a match event and links the bill's bill segment and adjustment FT's to it.
- Refer to the Bill ID Match Type Algorithm for more information about this algorithm.

NOTE:

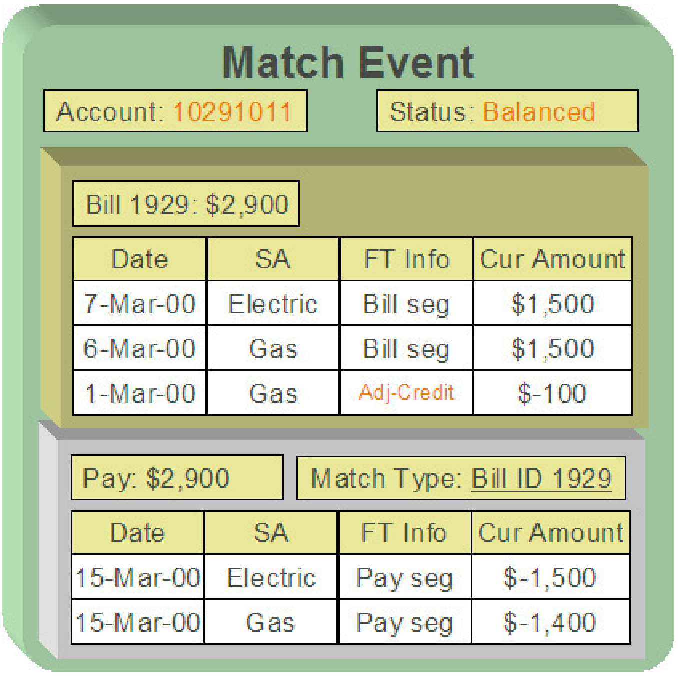
> **The match type's distribution logic is not "hard coded".** Because the match type's payment distribution logic is embedded in a plug-in algorithm, you can introduce new algorithms as per your company's requirements.

It's worth noting that payment *distribution* and *freezing* are two separate steps that typically happen in quick succession. The system's standard match event algorithms create the match event during payment distribution. This match event exists in the open state (because the payment segment's FT's have not yet been linked to the match event and therefore debit FT's do not equal credit FT's). The open match event references the debit FT's (the bill segments and adjustments) for which it pays. It is only at payment freeze time that the credit FT's (the payment segments) are linked to the match event thus allowing the match event to be become balanced.

If, at freeze time, the payment's credit FT's do not equal the debit FT's on the match event, the match event is left in the open state. An alert will appear on Control Central to highlight the existence of open match events (if the appropriate alert algorithm is plugged in the installation record). In addition, you can also set up a To Do entry to highlight the existence of open match events.

Payments Are Matched To Debit Credit FTs

While the above discussion dealt with the typical situation where the payment's credit FT's are matched against a bill's debit FT's, we want to note that a payment's FT's may be matched against debit and credit FT's. Consider the following example:



Match Event

Account: 10291011

Status: Balanced

Bill 1929: $2,900

Date	SA	FT Info	Cur Amount
7-Mar-00	Electric	Bill seg	$1,500
6-Mar-00	Gas	Bill seg	$1,500
1-Mar-00	Gas	Adj-Credit	$-100

Pay: $2,900

Match Type: Bill ID 1929

Date	SA	FT Info	Cur Amount
15-Mar-00	Electric	Pay seg	$-1,500
15-Mar-00	Gas	Pay seg	$-1,400

Notice that:

- The $2,900 payment is distributed amongst two service agreements (electricity and gas).
- The FT's to which the payment segments are matched are both debit and credit FT's. Notice that the debit FT's (the bill segments) and the credit FT (the adjustment) sum to $2,900.

Credits may result in a situation where the total amount on a bill for an SA is negative. This would be the case if in the above example the credit adjustment were for $-1600 resulting in the total amount for the Gas SA on this bill to be $-100 (credit). Assume a full payment of $1400 is made towards this bill. The Bill ID Match Type Algorithm first allocates negative payment amounts to any SA credit amount on the bill being paid. It then carries over the credit amount to pay off other bill amounts. In this example, a "negative" payment segment is created to match the $-100 credit of the Gas SA. Using the carried over credit a $1500 payment segment is created to match the $1500 debit of the Electric SA.

How Are Match Events Cancelled?

A user can cancel an open or balanced match event at any time. When a match event is cancelled, the event's FT's again effect arrears (and they can be associated with new match events). In other words, when a match event is cancelled, its FT's are released from the match event and become open-items.

In addition to manual cancellation, the system may automatically cancel a match event when the last of its payment FT's, if any, is cancelled (if you plug-in the appropriate FT freeze plug-in on your open-item customer classes).

For example, consider a match event that was created when a payment was made. If the payment is subsequently cancelled, the match event is also cancelled (thus releasing the match event's FT's) if no other payment FT's are linked to the match event. Please be aware that FT cancellation also causes a new match event to be created. This match event matches the original FT (the payment segment) and its cancellation FT. This means that the only "open items" that will exist after a payment is cancelled are the debit FT's that were originally paid.

➤

NOTE:

Reopening bills associated with automatic payment customers. While many payments are cancelled due to non-sufficient funds, please be aware that if you reopen a bill for which an automatic payment was created, the system will cancel the associated payment. If this payment is associated with a match event (because the account is an open-item account), the match event will be cancelled and a new match event will be created to match the original automatic payment with its cancellation details. This is necessary because a new payment will be created with the bill is subsequently completed and this payment's FT's will be matched to the bill's FT's.

Canceling a payment can result in many match events being created. If a cancelled payment has multiple payment segments, a separate match event will be created for each payment segment.

While payment cancellation is the most common type of FT cancellation, be aware that bill segment or adjustment cancellation may also cause a new match event to be created. We don't necessarily want to always link the cancellation FT and its original FT to the same match event. For example, when the cancellation FT is swept on to the next bill it affects the next bill and not the original FT's bill. For cancellations that will not be swept on to the next bill (payment cancellation, cancellation of an adjustment that is not shown on bill, credit notes, and bill segment cancellation before the bill is completed) the system creates a new match event that matches the original FT and its cancellation FT. This way, neither FT affects aged debt. If the original FT was linked to an existing match event and no other FTs are left on this match event it is automatically canceled.

Current Amount Is Matched, Not Payoff

The system matches the current amount of financial transactions, not the payoff amount.

●

FASTPATH:

Please refer to Current Amount versus Payoff Amount for more information about current and payoff amounts.

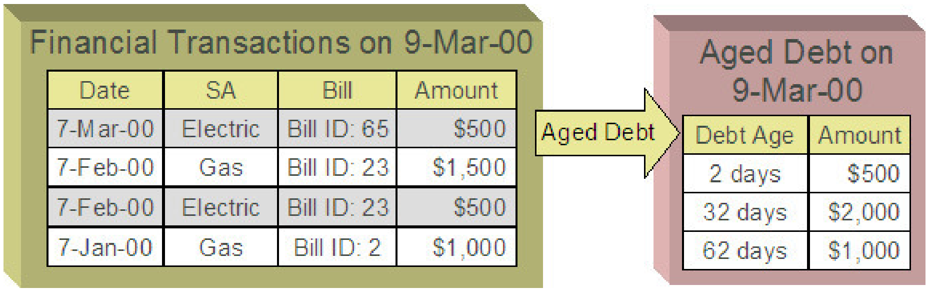
Disputing Items

Open-item customers may dispute FT's that they are not comfortable paying. For example, a customer who receives a bill with an anomalous charge may decide to dispute it.

When an open-item customer disputes a charge, a user creates a match event and links the disputed FT(s) to it. This match event will be in the open state (because it does not contain FT's that sum to zero). In addition, the match event's "disputed switch" is turned on.

NOTE:
Alerts. An alert is displayed on control central to highlight the existence of disputed match events (if the appropriate alert algorithm is plugged in). In addition, you can also set up a To Do entry to highlight the existence of disputed match events.

While the dispute is being researched, the disputed amount will not affect aged debt, but it still forms part of the customer's balance. For example, consider the following unpaid financial transactions that exist for an account:



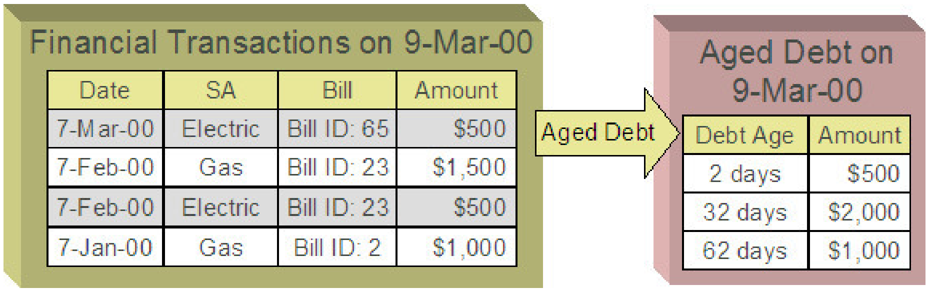
Financial Transactions on 9-Mar-00

Date	SA	Bill	Amount
7-Mar-00	Electric	Bill ID: 65	$500
7-Feb-00	Gas	Bill ID: 23	$1,500
7-Feb-00	Electric	Bill ID: 23	$500
7-Jan-00	Gas	Bill ID: 2	$1,000

Aged Debt on 9-Mar-00

Debt Age	Amount
2 days	$500
32 days	$2,000
62 days	$1,000

If the customer disputes the two electric bill segments, the customer's aged debt will look as follows:



Financial Transactions on 9-Mar-00

Date	SA	Bill	Amount
7-Mar-00	Electric	Bill ID: 65	$500
7-Feb-00	Gas	Bill ID: 23	$1,500
7-Feb-00	Electric	Bill ID: 23	$500
7-Jan-00	Gas	Bill ID: 2	$1,000

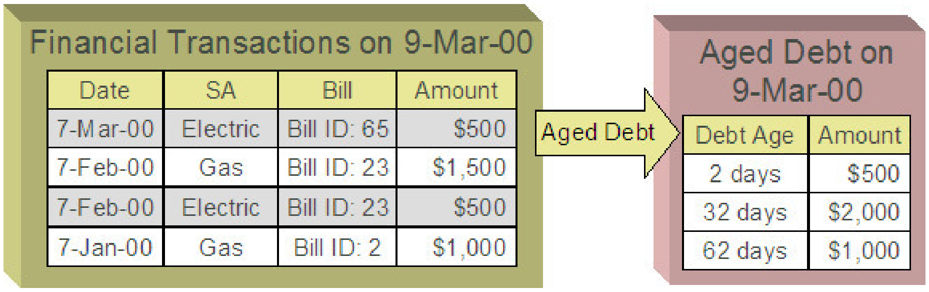
Aged Debt on 9-Mar-00

Debt Age	Amount
2 days	$500
32 days	$2,000
62 days	$1,000

Notice how a new category of debt appears - Disputed. Also notice how the 2 day old debt disappears and the 32 day old debt is reduced by the disputed amount.

The system shows disputed debt on Control Central. In addition, in all places where aged debt appears in the system, disputed debt is shown as a separate debt category.

If the dispute goes in your company's favor, the disputed match event should be cancelled (thus allowing the FT's to again impact aged debt). For example, if we assume 3 days have passed and the dispute match event is cancelled, the customer's aged debt will look as follows:



Financial Transactions on 9-Mar-00

Date	SA	Bill	Amount
7-Mar-00	Electric	Bill ID: 65	$500
7-Feb-00	Gas	Bill ID: 23	$1,500
7-Feb-00	Electric	Bill ID: 23	$500
7-Jan-00	Gas	Bill ID: 2	$1,000

Aged Debt on 9-Mar-00

Debt Age	Amount
2 days	$500
32 days	$2,000
62 days	$1,000

If the dispute goes in the customer's favor:

- You may decide to issue a credit note to cancel the offending bill or bill segment(s). As described above, the system in this case will automatically create new match events that match the original FTs with their cancellation FTs and cancel the disputed match event when the last item is unlinked from it.
- You may decide to cancel the offending bill segment(s) / adjustment(s). As described above, these cancellations are going to be swept on to the next bill. The system therefore will not automatically cancel the disputed match event. Notice that the cancellation effect of the disputed items is carried over on to the next bill. This means that the previously disputed items still need to be paid.

> **NOTE:**
> **Cancel / rebill.** If you cancel / rebill an offending bill segment, both the cancel and the rebill will become open-items that will be matched when the next bill is paid.

- You may decide to issue an adjustment to counter the effect of the disputed FT's. In this situation, you would simply link the adjustment FT to the disputed FT's (thus allowing the match event to become balanced). It is important to use in this case an adjustment that does not show on bill.

Pay Plans

You create a pay plan when a customer agrees to make one or more scheduled payments to satisfy past (or future) debt. These payments cannot be matched to open items because it is unlikely that debit FT's exist that equal the amount of each scheduled payment. However, you must specify a match type on all payments made by open-item customers. Therefore, a conundrum exists - the system requires a match type on payments made by open item accounts, but payments made for pay plans cannot be matched to existing FT's. This conundrum is solved by the fact that match types do not have to specify an override payment distribution algorithm. The customer class's standard distribution algorithm is used for payments that reference such a match type.

You may wonder how these payments will eventually get matched to open items? If ALL payments associated with a pay plan occur before the next bill is paid (or if the pay plan exists to satisfy future debt), these payments will be swept onto the match event that is created when the customer pays their next bill. However, if the pay plan exists to payoff historical debt and this debt has not been entirely paid by the time of the next bill, an unmatched event will exist when the customer pays their subsequent bills (if the payment amount doesn't match the amount of new charges on the bill). Why? Because, the customer is not paying the entire amount of the bill and therefore the system will not be able to match the payment to open items. If this occurs, we recommend canceling the match events that are created when the customer pays their subsequent bills. When the customer finally pays off all outstanding debt, the system will create a single match event that will contain all payments and bill segments.

Over Payments

If a customer overpays a bill (i.e., we receive more cash than receivables), we strongly recommend you set up the system to NOT keep the excess credit on the customer's regular service agreements. Rather, we recommend you segregate the receivable onto an "excess credit" service agreement. If you do this, the system will transfer any excess credits to the regular service agreements at bill completion time. When this transfer occurs, the same accounting described under Payments Segment Financial Transaction Algorithms Transfer Holding Amounts To Payable GL Accountsoccurs as shown in the following example. Note: this example assumes an excess credit of $110 was transferred to a normal service agreement and the normal service agreement had $10 of held payables.

FASTPATH:
Refer to Overpayment Segmentation for how to set up the system to segregate overpayments on a separate service agreement.

NOTE:
Why not keep excess credits on a customer's regular service agreement? Because the system can't differentiate between a credit that exists as a result of an overpayment and a credit that exists because of cancel/rebills, it would be impossible for the system to know if payables should be realized as a result of the reduced credit balance. However, if you keep overpayments on an excess credit service agreement, the system knows to treat any transference of these credits as "payments" and therefore it can transfer holding balances to true payables.

Event	Normal SA GL Accounting	Excess Credit SA GL Accounting
Bill segment created	A/R 110 Revenue <100> Tax Holding <10>	
Payment of $300 is received	Cash 110 A/R <110> Tax Holding 10 Tax Payable <10>	Cash 190 Overpay <190>
Bill segment created	A/R 110 Revenue <100> Tax Holding <10>	
Transfer excess credit amount to normal service agreement (when bill is completed).	Xfer 110 A/R <110>	Overpay 110 Xfer <110>
Because the transfer adjustment is the equivalent of a cash relief outstanding tax holding is relieved in proportion to the amount of receivables that are reduced by the transfer	Tax Holding 10 Tax Payable <10>	
Net effect of the transfer	Xfer 110	Overpay 110

	A/R <110>	Xfer <110>
	Tax Holding 10	
	Tax Payable <10>	

NOTE:

Prepaid taxes - future functionality. If your organization allows customers to prepay taxes in anticipation of a future tax increase (the customers receive the lower rate if they pay in advance), we do not consider this prepayment to be an overpayment. Rather, it is a payment of future taxes that will be remitted to the taxing authorities at payment time (due to cash accounting). Please speak to your account manager for when corresponding functionality will be available.

Setting Up The System To Enable Open Item Accounting

The following section provides an overview of how to enable open-item accounting.

Match Type Setup

The number of match types that you will need is dependent on the number of ways you want payments to be matched to open items. At a minimum, you will probably need the following match types:

- **Bill ID.** This match type should reference an override payment distribution algorithm that distributes the payment based on the bill ID specified on the payment (in match value). Refer to Payments And Match Events for more information.
- **SA ID.** This match type should reference an override payment distribution algorithm that distributes the payment based on the SA ID specified on the payment (in match value). Refer to Payments And Match Events for more information.
- **Pay Plan.** This match type should NOT reference an override payment distribution algorithm (if this algorithm is blank, the customer class's payment distribution algorithm is used). Refer to Pay Plans for more information.

Match Event Cancellation Reason Setup

The number of match event cancellation reasons that you will need is dependent on the number of ways your organization can justify the cancellation of a match event. At a minimum, you will probably need the following match event cancellation reasons:

- **FT Cancellation.** This cancel reason should be referenced on the Customer Class FT Freeze algorithm that is responsible for canceling match events when one of its financial transactions is cancelled.
- **Incorrect Allocation.** This cancel reason should be specified by users when they cancel match events that were created by the system erroneously.

Customer Class Setup

The following points describe customer class oriented set up functions:

- Turn on the open-item accounting switch.

- Set up the following algorithms for each CIS division:
    - Specify a **payment freeze** algorithm that causes a payment's FT's to be linked to the match event that was created when the payment was distributed. Refer to Payments And Match Events for more information.
    - Specify a **FT freeze** algorithm that causes match events to be cancelled (and a new match event to be created) when a FT is cancelled. Refer to How Are Match Events Cancelled for more information about cancellation.
    - We strongly recommend specifying an **overpayment** algorithm that causes overpayments to be segregated onto an "excess credit / overpayment" SA. Refer to Overpayments for more information.

Overpayment SA Type Setup

Specify a **bill completion** algorithm that causes the credit amount on overpayment SAs to be transferred to newly create debt (created when the bill is created). This algorithm transfers an overpayment SA's balance to regular SAs and creates a match event if the overpayment covers the entire bill. Refer to Overpayments for more information.

Installation Record Setup

Specify an **automatic payment** algorithm that causes a match event to be created when automatic payments are created for open-item accounts. The base package algorithm will do this for you if you specify the appropriate parameter on the algorithm. Refer to APAY-CREATE for more information about this algorithm.

If you want a Control Central alert to highlight when the current account has any open match events, plug in the appropriate **control central alert** algorithm on your installation record. Refer to C1-OPN-MEVT for more information about this algorithm.

If you want to enable manual pay segment distribution for open item accounts, along with other functions, you will need to plug in an installation algorithm for bill balance calculation. Refer to C1-OI-BI-AMT for more information about this algorithm.

To Do Entry Setup

Two To Do types are supplied with the base package:

- **TD-MODTL**. This To Do type highlights the presence of open, disputed match events.
- **TD-MONTL**. This To Do type highlights the presence of open, non-disputed match events.

Each of the above To Do types should be configured with the roles that work on entries of each type.

In addition, the account management group and/or divisions from which the default roles are extracted should be updated to define the role that should be defaulted for each of the above To Do types.

FASTPATH:
Refer to The Big Picture Of To Do Lists for more information about To Do lists.

Setting Up Match Types

Most payments are distributed amongst service agreements using the payment distribution algorithm specified on the payment's account's customer class. This algorithm decides how to distribute a payment amongst an account's existing debt if the customer doesn't specify how the payment should be distributed.

A customer can specify how a payment is distributed by specifying a match type and match value on their payments. Consider the following examples:

- Customers that are subject to open-item accounting (this is defined on the account's customer class) tell the system exactly which debt is covered by their payments. For example, an open-item customer might make a payment in respect of bill ID 123919101919.
- Even non open-item customers can direct payments to specific SAs. For example, the system allows a balance-forward customer's payment to be directed to a specific service agreement (however, they cannot direct payments to specific bills as only open-item customers can do this).

Match types are used to define the specific type of debt that is covered by a payment. The match type contains the algorithm that effectively overrides the standard payment distribution algorithm defined on the account's customer class.

> **NOTE:**
> **Background information.** Please refer to Payments And Match Events and Match Type Setup for more information about how match types are used.

To set up match types, select **Admin** > **Financial** > **Match Type.**

Description of Page

Enter an easily recognizable **Match Type** and **Description**.

Define the **Pay Dist Override Algorithm** used to distribute payments that reference this match type. If you haven't done so already, you must set up this algorithm in the system. To do this:

- Create a new algorithm (refer to Setting Up Algorithms).
- On this algorithm, reference an Algorithm Type that overrides the normal payment distribution algorithm.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference CI_MATCH_TYPE.

Setting Up Match Event Cancellation Reasons

When a match event is canceled, a cancel reason must be supplied.

> **NOTE:**
> **Background information.** Refer to How Are Match Events Cancelled? and Setting Up Match Event Cancellation for more information about cancellation.

To set up match event cancellation reasons, select **Admin** > **Financial** > **Match Event Cancel Reason.**

Description of Page

Enter an easily recognizable **Match Event Cancel Reason** and **Description**.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference CI_MEVT_CAN_RSN.

Fund Accounting

The topics in this section provide background information about fund accounting.

➤ **NOTE:**
This section is only relevant for some organizations. The system configuration requirements described in this section are only relevant if your organization practices fund accounting (this type of accounting is typically performed by municipal utilities). If your organization does not practice fund accounting, you need only indicate such on the Installation Record; no other setup is required.

Fund Accounting Overview

Municipal utilities, and not-for-profit organizations in general, often use a form of accounting different from that used by for-profit corporations. Municipal utilities typically practice fund accounting, whereas corporations practice corporate accounting.

Regulations or other restrictions may require a municipal utility to account for the finances of each of its departments as a separate entity. If a municipal utility provides both water and wastewater service, a municipal utility may need to track the receivables, revenue, and liabilities for water service separately from those of wastewater. In contrast, a corporation is free to co-mingle the moneys of the two services.

To track the services separately, the municipal utility sets up a fund for each department. A fund is an accounting entity with its own self-balancing set of accounts. Each fund has its own "sub general ledger" with its own chart of accounts, and within each fund, its debits equal its credits at all times. This allows the utility to report on the financial state of each fund independently.

In addition to having a fund for each department, there is also a general fund, which is used to handle inter-fund transfers as well as shared accounts.

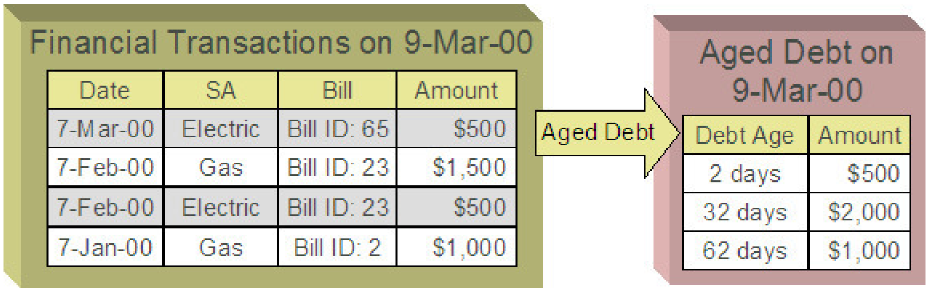
Fund Accounting Example

Consider a municipal utility which provides water and wastewater service. The utility has two departments: water and wastewater. Each department must track their finances separately therefore a fund is setup for each department:

- Water (fund 01).
- Wastewater (fund 02).

In addition, with fund accounting, there is always a general fund (fund 99).

Assume the following bill is generated.



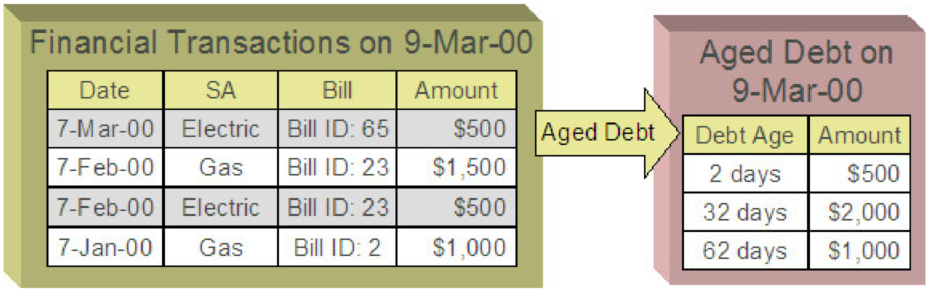
Financial Transactions on 9-Mar-00

Date	SA	Bill	Amount
7-Mar-00	Electric	Bill ID: 65	$500
7-Feb-00	Gas	Bill ID: 23	$1,500
7-Feb-00	Electric	Bill ID: 23	$500
7-Jan-00	Gas	Bill ID: 2	$1,000

Aged Debt on 9-Mar-00

Debt Age	Amount
2 days	$500
32 days	$2,000
62 days	$1,000

The bill would produce the following GL entries:



Financial Transactions on 9-Mar-00

Date	SA	Bill	Amount
7-Mar-00	Electric	Bill ID: 65	$500
7-Feb-00	Gas	Bill ID: 23	$1,500
7-Feb-00	Electric	Bill ID: 23	$500
7-Jan-00	Gas	Bill ID: 2	$1,000

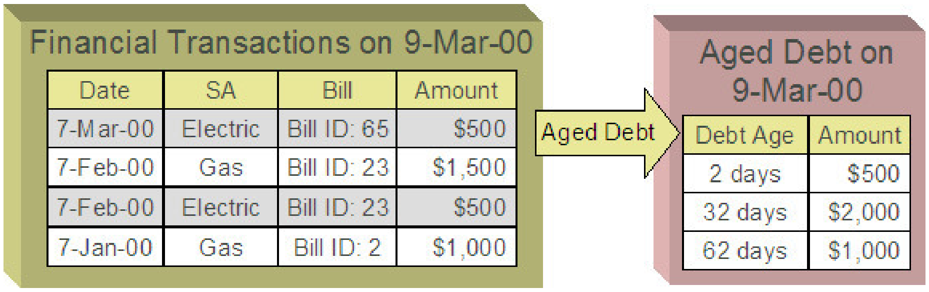
Aged Debt on 9-Mar-00

Debt Age	Amount
2 days	$500
32 days	$2,000
62 days	$1,000

For each fund, the GL details of the bill will include a debit to the accounts receivable (A/R) account and credits to the revenue and taxes payable accounts. In organizational terms, each department is owed a portion of the overall bill by the customer, part of which is sales by the department and part of which is owed to the taxing authorities by the department. Each fund is balanced.

Note that the accounting could be identical under corporate accounting if each service is its own division with its own chart of accounts.

The following diagram illustrates the initial GL accounting that would occur when the payment arrives:



Financial Transactions on 9-Mar-00

Date	SA	Bill	Amount
7-Mar-00	Electric	Bill ID: 65	$500
7-Feb-00	Gas	Bill ID: 23	$1,500
7-Feb-00	Electric	Bill ID: 23	$500
7-Jan-00	Gas	Bill ID: 2	$1,000

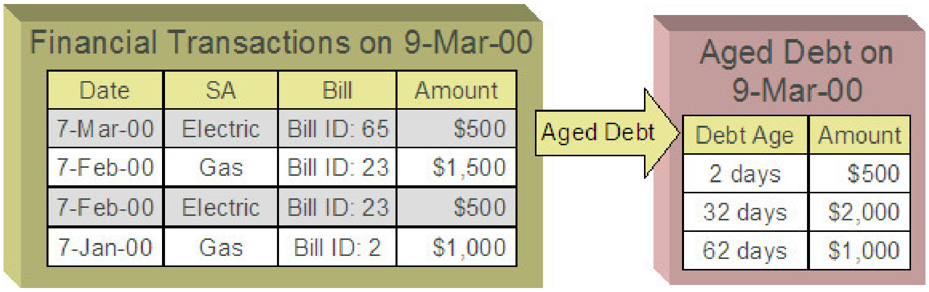
Aged Debt on 9-Mar-00

Debt Age	Amount
2 days	$500
32 days	$2,000
62 days	$1,000

The utility's general cash account is debited, and the departmental funds' A/R accounts are credited. In other words, the cash is held by the utility as a whole but the receivables are reduced for the individual departments.

If the accounting were left in this state, the fund accounting principal - that each fund represents an independent entity with a self-balancing chart of accounts - would be violated. This violation is caused due to the fact that cash is recorded on the general fund, not the departmental funds, causing the general fund to have an excess debit and the departmental funds to have an excess credit.

From an organizational viewpoint, to make each department whole, the departments need to note what portion of the cash they own, and correspondingly, the utility needs to note what portion of the cash is owed to each department. The following diagram illustrates this point.


Financial Transactions on 9-Mar-00

Date	SA	Bill	Amount
7-Mar-00	Electric	Bill ID: 65	$500
7-Feb-00	Gas	Bill ID: 23	$1,500
7-Feb-00	Electric	Bill ID: 23	$500
7-Jan-00	Gas	Bill ID: 2	$1,000

Aged Debt on 9-Mar-00

Debt Age	Amount
2 days	$500
32 days	$2,000
62 days	$1,000

To maintain a balance of debits and credits within each fund, the departmental funds have an "equity in pooled cash" (EPC) account and the general fund has a liability account for each departmental fund. In addition to debiting the general fund's cash account and crediting the departmental funds' A/R accounts, the departmental funds' EPC accounts are debited and the general funds liability accounts are credited.

And so, with the additional GL entries, all funds have matching debits and credits.

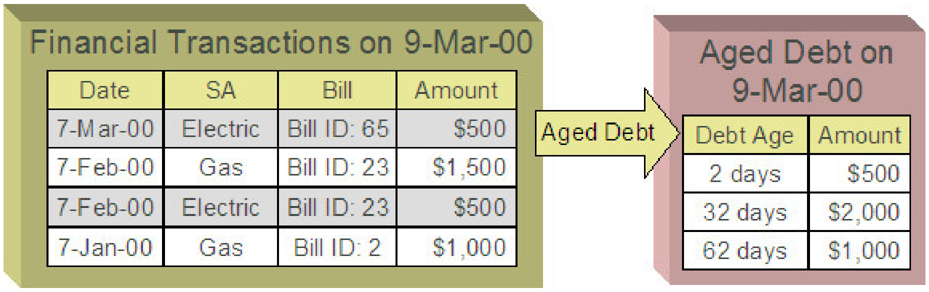
An Example Of A Bill Segment That References Multiple Funds

Consider a municipal utility that primarily supplies water service but is also responsible for maintaining the city's fire hydrants. The costs for fire hydrant maintenance are borne by the water customers and make up just a small portion of the overall bill. These costs are simply added to the water bill as a line item. The utility has two departments:

- Water service (fund 01)
- Hydrant maintenance (fund 39).

In addition, there is a general fund (fund 99).

Assume the following bill is generated for water and hydrant services.

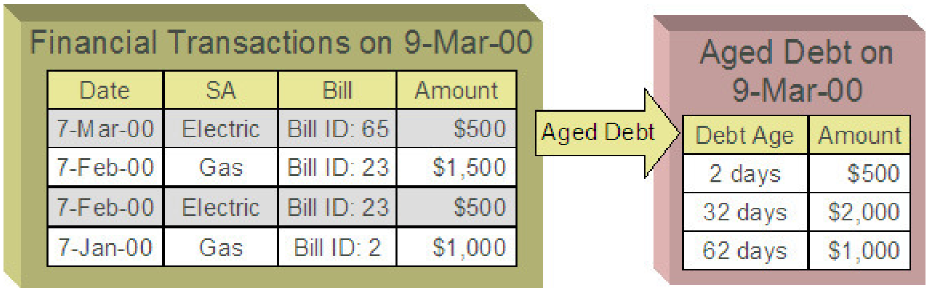

Financial Transactions on 9-Mar-00

Date	SA	Bill	Amount
7-Mar-00	Electric	Bill ID: 65	$500
7-Feb-00	Gas	Bill ID: 23	$1,500
7-Feb-00	Electric	Bill ID: 23	$500
7-Jan-00	Gas	Bill ID: 2	$1,000

Aged Debt on 9-Mar-00

Debt Age	Amount
2 days	$500
32 days	$2,000
62 days	$1,000

The following diagram illustrates the initial GL entries for the bill:


Financial Transactions on 9-Mar-00

Date	SA	Bill	Amount
7-Mar-00	Electric	Bill ID: 65	$500
7-Feb-00	Gas	Bill ID: 23	$1,500
7-Feb-00	Electric	Bill ID: 23	$500
7-Jan-00	Gas	Bill ID: 2	$1,000

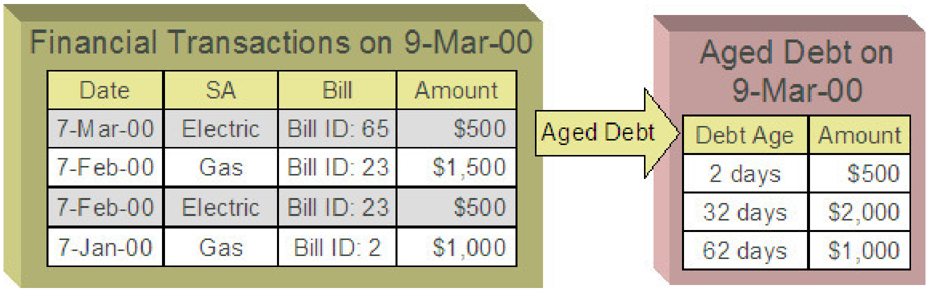
Aged Debt on 9-Mar-00

Debt Age	Amount
2 days	$500
32 days	$2,000
62 days	$1,000

In accounting for the bill, the water fund's A/R is debited, the water and hydrant funds' revenue accounts are credited, and the water's taxes payable account is credited.

If left at this, the funds would be out of balance; the water fund would have an overall excess debit and the hydrant fund would have an equal excess credit. In organizational terms, the hydrant fund has recorded sales but that amount is recorded as being owed to the water department.

To balance each department, the water department accepts the responsibility for collecting the hydrant charges from the customer but immediately remunerates the charges to the hydrant fund.


Financial Transactions on 9-Mar-00

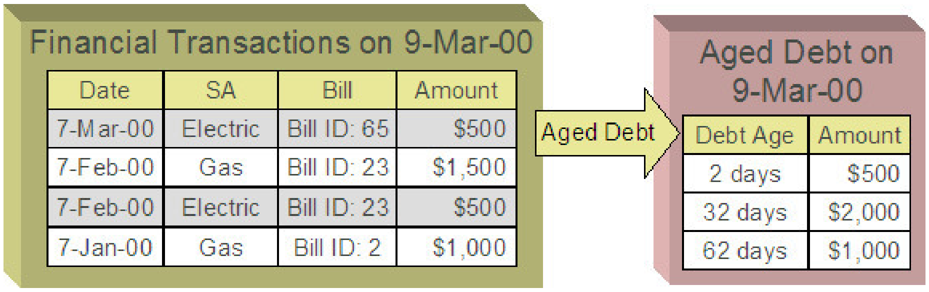
Date	SA	Bill	Amount
7-Mar-00	Electric	Bill ID: 65	$500
7-Feb-00	Gas	Bill ID: 23	$1,500
7-Feb-00	Electric	Bill ID: 23	$500
7-Jan-00	Gas	Bill ID: 2	$1,000

Aged Debt on 9-Mar-00

Debt Age	Amount
2 days	$500
32 days	$2,000
62 days	$1,000

This transfer is done using the general fund. The water fund's EPC account is credited and the liability to water is debited with the amount of the hydrant revenue. Also, the hydrant fund's EPC account is debited and the general fund's liability to hydrant account is credited by the hydrant revenue. In effect, the water department owes the hydrant charges to the general fund, and the general fund owes the hydrant charges to the hydrant fund.

The following diagram illustrates the initial GL accounting that would occur when the payment arrives:

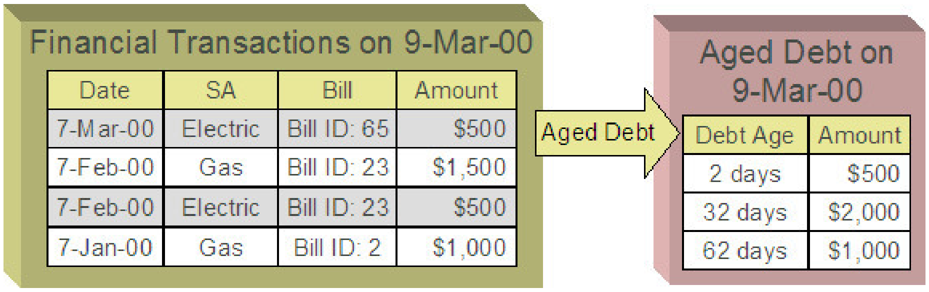

Financial Transactions on 9-Mar-00

Date	SA	Bill	Amount
7-Mar-00	Electric	Bill ID: 65	$500
7-Feb-00	Gas	Bill ID: 23	$1,500
7-Feb-00	Electric	Bill ID: 23	$500
7-Jan-00	Gas	Bill ID: 2	$1,000

Aged Debt on 9-Mar-00

Debt Age	Amount
2 days	$500
32 days	$2,000
62 days	$1,000

When the payment arrives, the cash is debited to the general fund's cash account, and the water fund's A/R is relieved. Again, the funds would be unbalanced if left in this condition; the water fund would have an excess of credits and the general fund would have an excess of debits.


Financial Transactions on 9-Mar-00

Date	SA	Bill	Amount
7-Mar-00	Electric	Bill ID: 65	$500
7-Feb-00	Gas	Bill ID: 23	$1,500
7-Feb-00	Electric	Bill ID: 23	$500
7-Jan-00	Gas	Bill ID: 2	$1,000

Aged Debt on 9-Mar-00

Debt Age	Amount
2 days	$500
32 days	$2,000
62 days	$1,000

To maintain each fund's balance of debits and credits, the general fund's liability to the water fund is credited by the amount of the department's share of the cash, and the water fund's EPC is debited. Note that the payment has no effect on hydrant fund's EPC and the general fund's liability to the hydrant fund. The hydrant department "received" its money from the water department when the bill was created.

And so, all funds have matching debits and credits.

Accounting Method Is Defined On The Installation Options

You must turn on a switch on the Installation Record to enable fund accounting.

Fund Controls Fund-Balancing Entries

There are two levels of debit and credit balancing in fund accounting. There is the balancing required by double entry accounting: the total debits in the entire GL must equal the total credits. This is required regardless of whether fund or corporate accounting is used. The distribution codes for these entries come from varying sources, depending on the type of financial event.

> **FASTPATH:**
> Refer to The Source Of GL Accounts On Financial Transactions for information on the sources of the distribution codes.

The second level of balancing is specific to fund accounting. Within each fund-not just across the GL-the total debits must equal the total credits. The original distribution code from the financial event has a fund specified. For example, a bill would cause a debit to a fund's A/R distribution code, and included in that A/R distribution code is the fund. It is the definition of the fund that specifies whether fund-balancing entries are required and provides the distribution codes for these entries.

For a departmental fund, the fund-balancing debit and credit would be specified. When a debit is applied to a departmental fund's GL account, an additional account (typically the general fund's liability to the departmental fund) is debited and an account (typically the departmental fund's EPC) is credited. When a credit is applied to a departmental fund's account, an additional account (typically the general fund's liability to the departmental fund) is credited and an account (typically the department's EPC) is debited.

For the general fund, no fund-balancing debits and credits are specified.

Building Fund-Balancing GL Details

Building the GL details for a financial event is a two-step process.

- First, the system generates the regular GL details for a financial transaction (FT). This is done regardless of whether corporate or fund accounting is used.
- Second, if fund accounting is activated (by turning on a switch on the Installation Record), the system analyzes the distribution code on each GL detail associated with the FT. If a fund is specified on a distribution code, the system checks the definition of the fund. If fund-balancing entries are specified on the fund, two additional GL entries are added to the FT:
- An offsetting entry to the Equity in Pooled Cash account is created for the departmental fund (e.g., if the FT is debiting a given fund, an offsetting credit is created in the funds EPC account).
- Another entry to the departments Liability account is created for the general fund.

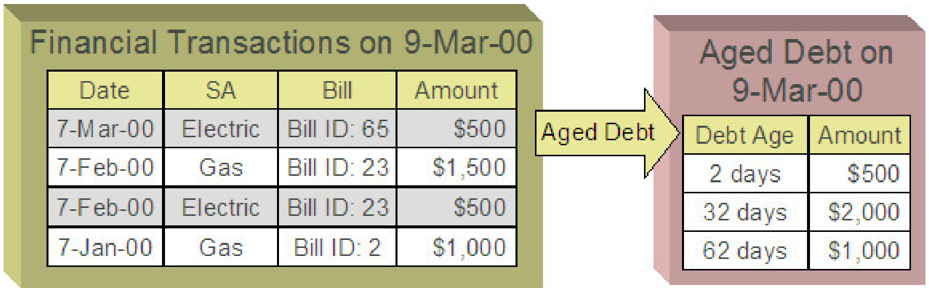
The result is a consolidated set of GL entries for the FT, incorporating the regular entries as well as the fund-balancing entries.

The topics in this section illustrate the generation of the GL details for the earlier examples.

FTs Whose GL Details All Reference The Same Fund Do Not Impact the General Fund or EPC Accounts

In Fund Accounting Example, where the bill's bill segments reference a single fund, the system creates a fund-balancing GL entry for each GL entry applied to a departmental fund:

- A debit to a departmental GL account triggers a debit to the general fund's liability-to-departmental-fund account and a credit to the departmental fund's equity-in-pending-cash account.
- A credit to a departmental GL account triggers a credit to the general fund's liability-to-departmental-fund account and a debit to the departmental fund's equity-in-cash account.



Financial Transactions on 9-Mar-00

Date	SA	Bill	Amount
7-Mar-00	Electric	Bill ID: 65	$500
7-Feb-00	Gas	Bill ID: 23	$1,500
7-Feb-00	Electric	Bill ID: 23	$500
7-Jan-00	Gas	Bill ID: 2	$1,000

Aged Debt on 9-Mar-00

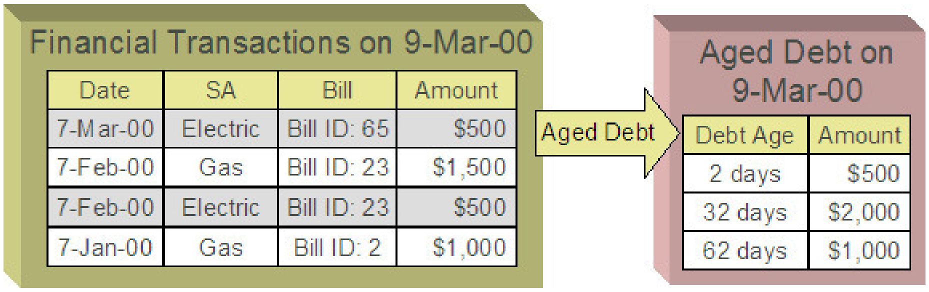
Debt Age	Amount
2 days	$500
32 days	$2,000
62 days	$1,000

The net effect on the individual equity-in-cash and general fund's liability accounts is zero because the debits and credits net to zero for each GL account. In other words, the yellow boxes net to zero and therefore fund accounting does not impact the bill segment's financial transactions. Refer to Fund Accounting Example for the resulting consolidated GL entries.

An FT Whose GL Details Reference Multiple Funds

In An Example Of A Bill Segment That References Multiple Funds, where the bill's bill segments reference multiple funds (water and hydrant), the system also creates fund-balancing GL entries for the financial transaction:

- A debit to a departmental GL account triggers a debit to the general fund's liability-to-departmental-fund account and a credit to the departmental fund's equity-in-pending-cash account.
- A credit to a departmental GL account triggers a credit to the general fund's liability-to-departmental-fund account and a debit to the departmental fund's equity-in-cash account.



Financial Transactions on 9-Mar-00

Date	SA	Bill	Amount
7-Mar-00	Electric	Bill ID: 65	$500
7-Feb-00	Gas	Bill ID: 23	$1,500
7-Feb-00	Electric	Bill ID: 23	$500
7-Jan-00	Gas	Bill ID: 2	$1,000

Aged Debt on 9-Mar-00

Debt Age	Amount
2 days	$500
32 days	$2,000
62 days	$1,000

The net effect of the bill on the GL is that the water fund's EPC has a credit of $0.80, the hydrant fund's EPC has a debit of $0.80, the general fund's liability to the water fund has a debit of $0.80, and the general fund's liability to the hydrant fund has a credit of $0.80. Note that, overall, the general fund's overall liability to the departmental funds nets to zero. Refer to An Example Of A Bill Segment That References Multiple Funds for the resulting consolidated GL entries.

Setting Up The System To Enable Fund Accounting

The following section provides an overview of how to enable fund accounting.

Turn On Fund Accounting

On the Installation Record, indicate that fund accounting is Practiced.

Defining Funds

A fund must be setup for each specific fund in your organization. Don't forget to also set up a fund for the general fund. Navigate using **Admin** > **Financial** > **Fund.**

Description of Page

Enter a **Fund** and a **Description** to identify the fund.

If this fund is used to balance other funds or to hold cash, indicate a **Fund Type** of General, otherwise indicate that it is Specific.

If the fund type is Specific, specify the **Equity Distribution Code** and **Liability Distribution Code**. These codes are used to balance financial transactions that span funds. The equity distribution code should belong to the same **Fund** as the one you are defining. The liability distribution code should belong to the general fund.

Distribution Codes Must Include Fund ID

All of your distribution codes must include their respective fund ID.

- **FASTPATH:**
  For more information, refer to Setting Up Distribution Codes.

Update Your Funds With Their Respective Equity and Liability Distribution Codes

After distribution codes have been setup, you must update your funds to indicate the equity and liability accounts used to balance inter-fund financial transactions.

United Kingdom VAT and CCL

The topics in this section provide information about value added tax (VAT) and climate change levy (CCL) charges that are specific to non-domestic customers in the United Kingdom market.

➤ **NOTE:**
Applicable for UK market. This section is only relevant for the United Kingdom market. Other markets may disregard this section.

UK VAT Overview

Two rates of VAT, referred to as standard rate VAT and reduced rate VAT, are applicable to energy related charges in the UK. Domestic (i.e., residential) customers always pay VAT at the reduced rate. Non-domestic customers on the other hand normally pay standard rate VAT. However, part or all of the energy related charges for a non-domestic customer might be subject to reduced rate VAT. De minimis and VAT declarations affect the percentage of the energy related charges that is subject to each VAT rate:

- If average daily usage at a premise for a given service type does not exceed a certain threshold (the de minimis limit), all energy related charges at that premise / service type are taxed at the reduced rate.
- Some non-domestic customers, such as those with a mixed-use premise, may be eligible to pay reduced rate VAT on part or all of their energy related charges. Customers may file a VAT declaration specifying the percentage of their energy related charges that are eligible for reduced rate VAT - refer to Maintaining Declarations for more information. If the declared percentage exceeds a given threshold, the declared percentage is deemed to be 100% and the customer pays VAT at the reduced rate on all energy related charges at that premise. VAT declarations are non-transferable and must be filed for each account, premise, and service type combination.

In addition to the potential for different rates of VAT to be applicable on a bill, UK tax regulations require that excess credits are considered a prepayment of energy related charges together with VAT. Refer to Excess Credits and UK VAT for more information.

The system comes supplied with various algorithms types that can be used to perform the VAT calculations.

UK CCL Overview

The climate change levy (CCL) is based on the amount of energy used that is subject to standard rate VAT. Similarly to VAT declarations, customers may file for exemption from CCL. A CCL declaration specifies the percentage of the CCL charges that the customer is exempt from - refer to Maintaining Declarations for more information. CCL declarations are non-transferable and must be filed for each account, premise, and service type combination.

CCL charges themselves are subject to standard rate VAT.

UK VAT and CCL Bill Examples

The following examples show how a bill for a non-domestic customer is affected by de minimis, VAT declarations, and CCL declarations. In the examples, standard rate VAT of 17.5%, reduced rate VAT of 5%, a VAT declaration threshold of 60%, and a CCL charge of 0.43p per unit of energy are used.

Example 1 - Normal Account

This example shows the bill for a normal non-domestic account with no declarations and consumption above the de minimis limit. Standard rate VAT is applied to all energy related charges (the standing charge and the per unit charge) and to the CCL charge.

Bill Line	Amount
Standing Charge	10.00
2,000 units @ 10p	200.00
CCL on 2,000 units @ 0.43p	8.60
VAT @ 17.5% on 218.60	38.25
Total	256.85

Example 2 - Account with Consumption Under the De Minimis Limit

In this example, consumption does not exceed the de minimis limit and therefore reduced rate VAT is applied to all the energy related charges. There is no climate change levy because only units subject to standard rate VAT are subject to CCL.

Bill Line	Amount
Standing Charge	10.00
2,000 units @ 10p	200.00
VAT @ 5% on £210.00	10.05
Total	220.05

Example 3 - Account with VAT Declaration

This example is for a non-domestic account with a VAT declaration of 20%. Consumption is above the de minimis limit. CCL applies to only 80% of the total units as 20% is subject to reduced rate VAT and therefore exempt. 80% of the energy related charges (the standing charge and the per unit charge) and all of the CCL charge are subject to standard rate VAT. 20% of the energy related charges are subject to reduced rate VAT.

Bill Line	Amount
Standing Charge	10.00
2,000 units @ 10p	200.00
CCL on 1,600 units @ 0.43p	6.88
VAT @ 17.5% on £174.88	30.60
VAT @ 5% on £42.00	2.10
Total	249.58

Example 4 - Account with VAT Declaration and CCL Declaration

This example is for a non-domestic account with a VAT declaration of 20% and a CCL declaration of 10%. Consumption is above the de minimis limit. CCL applies to only 80% of the total units as 20% is subject to reduced rate VAT and therefore exempt. The customer gets a credit (CCL relief) for 10% of the CCL charges as a result of the CCL declaration. 80% of the energy related charges (the standing charge and the per unit charge) and all of the CCL charge less the CCL reliefs are subject to standard rate VAT. 20% of the energy related charges are subject to reduced rate VAT.

Bill Line	Amount
Standing Charge	10.00
2,000 units @ 10p	200.00
CCL on 1,600 units @ 0.43p	6.88
CCL relief	<0.68>
VAT @ 17.5% on £174.20	30.48
VAT @ 5% on £42.00	2.10
Total	248.78

Billing and UK VAT

The following sections describe how VAT rules are implemented when a bill is produced.

Application of De Minimis

The de minimis rule specifies that all energy related charges at a premise be subject to reduced rate VAT if the total usage at a premise does not exceed a certain threshold. To determine if de minimis applies, the total billed consumption for a premise must therefore be known. However, energy usage at a premise may be measured with several meters and billed by multiple service agreements. To ensure that the final billed consumption for the premise is used to determine if de minimis applies, all service agreements for an account of a given service type at a given premise should be billed together and de minimis should be checked after the bill segments have been generated. In addition, all bill segments on one bill for consumption at a given premise must be for the same bill period.

➤ **NOTE:**
Bill segments from different periods may not appear on the same bill. If a bill segment is canceled, all bill segments associated with consumption at a given premise for the same service type must be canceled and re-billed together. You cannot re-bill the bill segment on a bill for a different period.

An algorithm type is supplied with the base package that checks for de minimis at bill completion time. The base bill completion algorithm type checks that for any service agreement that was billed, bill segments for all the account's service agreements with the same service type and characteristic premise exist on the bill. It then calculates the total consumption for the premise and determines if de minimis applies. If de minimis applies, it sets a bill characteristic whose value is the premise ID to indicate that de minimis applies and regenerates the bill segment. Refer to the algorithm type CPBC-DMCH for more information on how this type of algorithm operates.

The calculation rule calculation algorithm type that calculates standard and reduced rate VAT (RCAM-VAT) applies the de minimis rule if the bill characteristic for de minimis is found. Refer to Calculation of VAT for more information. Note that during the initial generation of each bill segment, the characteristic will not exist and standard rate VAT will be applied. This means that if you look at a bill before it is complete, VAT may not be accurately reflected.

> **NOTE:**
> **Batch billing cannot regenerate re-billed bill segments.** If you cancel and rebill a bill segment, you should complete the bill on-line so that the re-billed bill segments can be regenerated if de minimis applies.

Calculation of VAT

To calculate VAT, the percentages of energy related charges subject to standard rate VAT and reduced rate VAT must be determined. As these percentages vary from customer to customer and even from one bill to the next for the same customer, their calculation must take place at billing time and is handled by a calculation rule calculation algorithm.

A calculation rule calculation algorithm type RCAM-VAT is provided to calculate the percentages of energy related charges subject to standard rate and reduced rate VAT, taking into account de minimis and any VAT declaration that is in effect for the service type, account, and premise. The percentages can then be applied to the appropriate charges that are cross-referenced.

Calculation of CCL

CCL is a charge per unit of energy subject to standard rate VAT. It is therefore dependent on the same percentage of energy related charges subject to standard rate VAT determined during the calculation of VAT. A calculation rule calculation algorithm handles calculation of CCL.

A calculation rule calculation algorithm type RCAM-CCL is provided to calculate CCL charges and CCL relief, taking into account any CCL declarations that are in effect for the service type, account, and premise.

Excess Credits and UK VAT

When a financial transaction that results in a credit balance for a service agreement is frozen, the amount of the excess credit must be accounted for as a prepayment of energy related charges and VAT. VAT is calculated at the reduced rate for domestic customers and at the standard rate for non-domestic customers. When the excess credit is used, the VAT liability is reversed.

A customer class FT freeze algorithm type CFTZ-VAT-GL is provided to create additional GL detail entries for unbilled revenue and VAT liability when the freezing of an FT results in a credit balance or a change to a service agreement's credit balance.

Excess Credit GL Accounting Example

The following example shows the additional GL details that are created when a service agreement's balance changes and the starting or ending balance is a credit. For this example, a single VAT rate of 10% is used simply to illustrate the principle.

The service agreement has a zero starting balance. Note the following:

- When the first payment is received, the overpayment results in a credit balance and additional GL entries are created to recognize the unbilled energy revenue and unbilled VAT liability.

- When the second bill segment is created, the credit balance is reduced and additional GL entries are created to back out the unbilled GL entries, up to the amount of the credit balance.
- When the bill segment is canceled, the service agreement again has a credit balance and additional GL entries are created to recognize the unbilled energy revenue and unbilled VAT liability.

Event	Normal GL Accounting	Additional GL Accounting	SA Balance
Bill segment for £110 created	A/R 110 Revenue <100> VAT <10>		110
Payment of £330 is received	Cash 330 A/R <330>	A/R 220 Unbilled Energy <200> Unbilled VAT <20>	<220>
Bill segment for £275 created	A/R 275 Revenue <250> VAT <25>	A/R <220> Unbilled Energy 200 Unbilled VAT 20	55
Payment of £55 is received	Cash 55 A/R <55>		0
Bill segment for £275 is canceled	A/R <275> Revenue 250 VAT 25	A/R 275 Unbilled Energy <250> Unbilled VAT <25>	<275>

Setting Up The System For UK VAT and CCL

The following sections provide an overview of how to configure the system to apply UK VAT and CCL charges for non-domestic customers.

Bill Segment Freeze Installation Option

Select Freeze at Bill Completion as the bill segment freeze option. The system determines if de minimis applies at bill completion time and has to regenerate the bill segments as necessary.

Customer Class

Create a separate customer classes for non-domestic customers.

Service Agreements

You must configure the system so that each service agreement that must be checked for de minimis is only linked to service points at a single premise. The service agreement must reference that premise as its characteristic premise. Refer to Application of De Minimis for more information.

Bill Factors for UK VAT and CCL

- Standard Rate VAT
- Reduced Rate VAT
- VAT Declaration Percentage Threshold. Set up a separate bill factor for each service type as the thresholds differ based on service type.
- De Minimis Average Daily Amount Threshold. Set up a separate bill factor for each service type as the de minimis amounts differ based on service type.
- Climate Change Levy (CCL) Per Unit Price. Set up a separate bill factor for each service type as the CCL per unit prices differ based on service type.

Characteristic Type for De Minimis Amounts

Set up a characteristic type to use for the bill characteristic that indicates when de minimis applies. Define the type of characteristic value as a Foreign Key Value and specify a foreign key reference that points to the premise table. Specify this characteristic type as parameters to the algorithms that apply the de minimis rule and that apply VAT.

You need to define a characteristic type for every service type that is subject to the de minimis rule, as the de minimis limit may apply for one service type but not another on the same bill.

Distribution Code

Define the following distribution codes:

- Unbilled Prepaid Energy Related Charges
- Unbilled Prepaid VAT

These distribution codes are required as parameters to the algorithms to create the GL details for excess credit.

UOM / TOU / SQI

The algorithm that calculates CCL requires the following service quantities:

- The percentage of energy related charges that is subject to standard rate VAT
- Define the SQI used to store this service quantity. You will specify this SQI as a parameter to the algorithm used to calculate VAT. Refer to the RCAM-VAT algorithm type for more information about the base package algorithm.
- The total consumption
- Define the UOM / TOU / SQI used to store these service quantities.

Service Quantities

The algorithms that calculate VAT and CCL require a service quantity containing the total consumption. Unless you have registers that measure the total consumption independent of time of use or interval, you will need to set up a pre-processing calculation rule or rate calculation algorithm to aggregate the usage into one service quantity.

Algorithms for UK VAT and CCL

Add the following algorithms:

- Apply De Minimis Rule. Define an algorithm for each service type for which de minimis should be checked. Plug the algorithm(s) in on the customer class for the Pre Bill Completion system event. Refer to the CPBC-DMCH algorithm type for more information about the base package algorithm.
- Apply VAT to Cross-Referenced Calculation Rules. Define one algorithm to calculate VAT at the standard rate and one algorithm to calculate VAT at the reduced rate (specify whether to calculate standard rate or reduced rate VAT using the algorithm parameter). You will need to define a set of algorithms for each service type (the bill characteristic to store the premise when de minimis applies and the VAT Declaration percentage threshold bill factor referenced in the algorithm parameters are different for each service type). Plug these algorithms in on the calculation rule for the Calculation Algorithm system event. Refer to the RCAM-VAT algorithm type for more information about the base package algorithm.
- Create Excess Credit GL Details. Plug this algorithm in on the customer class for the FT Freeze system event. Refer to the CFTZ-VAT-GL algorithm type for more information about the base package algorithm.
- Calculate CCL. Plug this algorithm in on the calculation rule for the Calculation Algorithm system event. Refer to the RCAM-CCL algorithm type for more information about the base package algorithm.
- Highlight Effective Declarations for Account and Premise. Plug this algorithm in on the installation option for the Control Central Alert system event. Refer to the CCAL-DECL algorithm type for more information about the base package algorithm.

Calculation Rules to Charge VAT and CCL

Four additional calculation rules are required to charge for VAT and CCL on non-domestic rates:

- Calculate standard rate VAT
  - Calculation rule type = Calculation Algorithm
  - Value type = Percentage

It is recommended that you use a value source of Bill Factor and reference the bill factor you set up for standard rate VAT.

- Specify the UOM/TOU/SQI of the service quantity that holds the total consumption.
- Turn on Derive SQ
- Calculation algorithm = the algorithm you set up to calculate VAT at the standard rate
- Cross-reference all calculation rules that contribute to the total bill amount for energy related charges
- Note that the algorithm overrides description on bill
- Calculate reduced rate VAT
  - Calculation rule type = Calculation Algorithm
  - Value type = Percentage

It is recommended that you use a value source of Bill Factor and reference the bill factor you set up for reduced rate VAT.

- Specify the UOM/TOU/SQI of the service quantity that holds the total consumption.

- Turn on Derive SQ
- Calculation algorithm = the algorithm you set up to calculate VAT at the reduced rate
- Cross-reference all calculation rules that contribute to the total bill amount for energy related charges
- Note that the algorithm overrides description on bill
- Calculate CCL and CCL Relief
  - Calculation rule type = Calculation Algorithm
  - Value type = Unit Rate

It is recommended that you use a value source of Bill Factor and reference the bill factor you set up for CCL per unit price.

- UOM/TOU/SQI = the identifier of the service quantity containing the total units of energy consumed
- Calculation algorithm = the algorithm you set up to calculate CCL
- Note that the algorithm overrides description on bill
- VAT on CCL
  - Calculation rule type = Apply To
  - Value type = Percentage

It is recommended that you use a value source of Bill Factor and reference the bill factor you set up for standard rate VAT.

- Cross-reference the calculation rule that calculates CCL and CCL Relief

Setting up the calculation rules as above produces separate lines for VAT on energy related charges and on CCL (i.e., lines will be created for standard rate VAT on energy related charges, reduced rate VAT on energy related charges, and standard rate VAT on CCL charges).

You can set up rates to calculate standard rate VAT on energy related charges and CCL together as follows:

- Calculate standard rate VAT charges
  - Calculation rule type = Calculation Algorithm
  - Turn on For Calculation Purposes Only (Result Type = Charge)
  - Value type = Percentage

Define a value source of Value and a value of 100.

- Specify the UOM/TOU/SQI of the service quantity that holds the total consumption.
- Turn on Derive SQ
- Calculation algorithm = the algorithm you set up to calculate VAT at the standard rate
- Cross-reference all calculation rule that contribute to the total bill amount for energy related charges.
- Calculate CCL and CCL Relief
  - Calculation rule type = Calculation Algorithm
  - Value type = Unit Rate

It is recommended that you use a value source of Bill Factor and reference the bill factor you set up for CCL per unit price.

- UOM/TOU/SQI = the identifier of the service quantity containing the total units of energy consumed
- Calculation algorithm = the algorithm you set up to calculate CCL
- Note that the algorithm overrides description on bill.
- Calculate standard rate VAT
  - Calculation rule type = Apply To

  - Value type = Percentage

It is recommended that you use a value source of Bill Factor and reference the bill factor you set up for standard rate VAT.

- Cross-reference the calculation rule that calculates standard rate VAT charges and the calculation rule that calculates CCL and CCL Relief
- Calculate reduced rate VAT:
  - Calculation rule type = Calculation Algorithm
  - Value type = Percentage

It is recommended that you use a value source of Bill Factor and reference the bill factor you set up for reduced rate VAT.

- Specify the UOM/TOU/SQI of the service quantity that holds the total consumption.
- Turn on Derive SQ
- Calculation algorithm = the algorithm you set up to calculate VAT at the reduced rate
- Cross-reference all calculation rules that contribute to the total bill amount for energy related charges.
- Note that the algorithm overrides description on bill.

Bill Taxation Threshold

Some implementations only apply taxes if the accumulated tax amount at the bill level exceeds some specified threshold amount.

Taxation Threshold Examples

The following examples show how taxation thresholds affect a customer's bill. In the examples a tax rate of 5% and a threshold amount of $21.30 is used.

Example 1 - Account With Taxes Under Threshold

This example shows the bill for an account where the accumulated tax amount is less than the threshold amount. Since the accumulated tax amount of $10.50 is less than the threshold amount, taxes are not applicable and the account's bill should be adjusted to exclude the tax amount of $10.50.

Bill Line	Amount
Standing Charge	10.00
2,000 units @ $0.10	200.00
Tax @ 5% on $210.00	10.50
Total	220.50
Adjusted Total	210.00

Example 2 - Account With Taxes Above Threshold

This example shows the bill for an account where the accumulated tax amount is greater than the threshold amount. Since the accumulated tax amount of $27.92 is greater than the threshold amount, taxes are applicable and the account should be billed for the total amount.

Bill Line	Amount
Standing Charge	10.00
4,000 units @ $0.10	400.00
Additional Charge	148.45
Tax @ 5% on $558.45	27.92
Total	586.37

Example 3 - Account With Rounding Discrepancy

This example shows the resulting bill segment calc lines for an account with 3 service agreements. Here the accumulated tax amount at the bill level is $21.29 with taxes calculated and rounded for each SA's bill segment. However, if the same taxes were calculated for each SA's bill segment and then accumulated and rounded at the bill level, the accumulated tax amount would be $21.30 implying that taxes are applicable and the account should be billed for the tax amount of $21.30 accounting for the discrepancy of $0.01.

Bill Segment	Bill Line	Amount	Pre-rounding Amount
Bill segment for SA 1	964.70 units @ $0.10	96.47	96.47000
	Tax @ 5% on $96.70	4.82	4.82350
	Bill Segment Total	101.29	
Bill segment for SA 2	2222.90 units @ $0.10	222.29	222.29000
	Tax @ 5% on $222.29	11.11	11.11450
	Bill Segment Total	233.40	
Bill segment for SA 3	1072.40 units @ $0.10	107.24	107.24000
	Tax @ 5% on $107.24	5.36	5.36200
	Tax Discrepancy	0.01	
	Bill Segment Total	112.61	

➤ **NOTE:**

Pre-rounding amount. Rate application captures two calculated amounts on the resulting bill segment calc lines. The first is the calculated amount rounded to two decimal places, and the second is a raw calculated amount with a five decimal precision. The base package algorithm that calculates taxation thresholds uses both amounts to account for any rounding discrepancy; however, only the raw calculated amount is used to compare against the taxation threshold.

Billing and Taxation Thresholds

The following sections describe how taxation threshold rules are implemented when a bill is produced.

Calculation of Taxation Thresholds

When taxation thresholds are applied at the account's bill level, it means that the system must calculate taxes for each of the account's service agreements, then sum these tax amounts and apply any applicable rounding rules. This accumulated tax amount is compared to a threshold amount and if the accumulated tax amount is less than the threshold amount, then taxes should not be applied to the customer's bill. To ensure the accumulated tax amount is accurate, all service agreements for an account whose tax amounts should be taken into consideration when comparing to the specified threshold should be billed together and the threshold comparison should take place after the bill segments have been generated. In addition, all bill segments whose tax amounts should be taken into consideration when comparing to the specified threshold must be for the same bill period.

➤ NOTE:
Bill segments from different periods may not appear on the same bill. If a bill segment is canceled, all bill segments associated with that bill must be canceled and re-billed together. You cannot re-bill the bill segment on a bill for a different period.

An algorithm type is supplied with the base package that checks for taxation thresholds at bill completion time. The base bill completion algorithm type accumulates identified tax calc line amounts (accomplished using a bill segment calc line characteristic), and compares this to a specified threshold amount to determine if taxes apply for the account. If taxes do not apply, it sets a bill characteristic indicating this and regenerates the bill segment. Refer to the algorithm type C1-CPBC-TAXT for more information on how this type of algorithm operates.

➤ NOTE:
Calculated adjustments are included in the evaluation against taxation thresholds. Adjustments that use a rate to calculate the adjustment amount may be included in the taxation threshold evaluation if their rate's components are set up to do so. The base algorithm C1-CPBC-TAXT looks at adjustments that are about to be swept onto the bill and, if applicable, includes them in the calculation.

The calculation rule(s) that calculate taxes make use of calculation rule eligibility criteria to ensure that taxes are only computed if the bill does not have the characteristic indicating that taxes are not applicable. Note that during the initial generation of each bill segment, the characteristic will not exist and taxes will be applied. This means that if you look at a bill before it is complete, taxes may not be accurately reflected.

Tax Amount Discrepancies

Since tax calculation rules are calculated and rounded at the bill segment level, it's possible that rounding discrepancies may occur if rounding of these tax amounts occurs at the bill level instead as illustrated in the example above. To account for this, the system uses both the two decimal precision and the five decimal precision calculated amounts that rate application captures on bill segment calc lines; however, only the raw calculated amount is used to compare against the taxation threshold. If there is a discrepancy in the tax amount (as shown in example 3 above), the system captures this amount as an entry in one of the bill segment's SQ collections prior to regenerating the bill segments. This is depicted in example 3 above where the tax discrepancy SQ resulted in an additional bill segment calc line on one of the bill segments for the rounding amount of 0.01. Note that a calculation rule is configured to bill for this discrepancy amount SQI.

> **NOTE:**
> **Pre-rounding amount.** Rate application captures two calculated amounts on the resulting bill segment calc lines. The first is the calculated amount rounded to two decimal places, and the second is a raw calculated amount with a five decimal precision. The base package algorithm that calculates taxation thresholds uses both amounts to account for any rounding discrepancy; however, only the raw calculated amount is used to compare against the taxation threshold.

Setting Up The System For Bill Taxation Thresholds

This section provides an overview of how to configure the system to calculate taxes at the account's bill level.

Installation Option

Select Freeze at Bill Completion as the bill segment freeze option. The system compares tax amounts calculated to a specified threshold amount and based on this determines if taxes should apply at bill completion time. If taxes should not be applied for the account, the system has to regenerate the bill segments as necessary.

Adjustment Types

Select Freeze at Bill Completion as the adjustment freeze option. The system compares tax amounts calculated to a specified threshold amount and based on this determines if taxes should apply at bill completion time. If taxes should not be applied for the account, the system has to regenerate the adjustments as necessary.

Bill Factors

Tax Threshold. Set up a separate bill factor for each distinct tax threshold amount.

Characteristic Type

Set up a characteristic type and value to identify the tax calculation rules and bill segment calculation lines that the system will use to compare to the specified threshold amount. Specify this characteristic type and value as parameters to the algorithms that apply the taxation threshold. This characteristic type and value must also be specified on each of your tax calculation rules that should be included in the threshold comparison.

Set up a characteristic type and value to use for the bill characteristic that indicates when bill level taxes apply. Specify this characteristic type and value as parameters to the algorithms that apply the taxation threshold.

Service Quantity Identifiers

Optional service quantity identifiers may be configured to capture the following:

- Tax Amount. The system compares the accumulated bill's tax amount to the specified threshold amount to determine if taxes are applicable. If taxes should not be applied for the account, the system regenerates the bill segments as necessary. If you wish to capture the tax amount computed for informational purposes, then a tax amount SQI should be set up and specified as a parameter on the algorithms that apply the taxation threshold.
- Tax Discrepancy Amount. Since tax calculation rules are calculated and rounded at the bill segment level, it's possible that rounding discrepancies may occur if rounding of these tax amounts should take place at the bill level. If you wish to account for these rounding discrepancies, then a tax discrepancy amount SQI should be set up and specified as a parameter on the algorithms that apply the taxation threshold. Your rates should also be configured to cater for this rounding discrepancy. The system adds an entry for the discrepancy amount to one of the bill segment's SQ collections prior to regenerating the bill segments.

Algorithms

Apply Taxation Threshold. Define an algorithm for each distinct tax threshold amount. Plug the algorithm(s) in on the appropriate customer class for the Pre Bill Completion system event. Refer to the C1-CPBC-TAXT algorithm type for more information about the base package algorithm.

Adjustment Generation - Apply Rate. Define an algorithm for each rate to be used by calculated adjustment types that are to be included in the tax threshold evaluation. Plug the algorithm(s) in on the appropriate adjustment type for the Generate Adjustment system event. Refer to the ADJG-RT algorithm type for more information about the base package algorithm.

Calculation Rules For Bill Taxation Thresholds

Your calculation rules that bill for taxes require the following:

- A characteristic that identifies them as calculation rules to include in taxation threshold comparisons
- Calculation rule eligibility criteria to ensure that taxes are not calculated if the taxation threshold algorithm dictates this

Only one eligibility group on the calculation rule is required. It would look as follows:

Group No.	Group Description	If Group is True	If Group is False
1	Tax applies if total accumulated tax amount at the account's bill level exceeds the threshold amount	Apply calculation rule	Skip calculation rule

The following criteria will be required for this group:

Seq	Field to Compare	Comparison Method	If True	If False	If Insufficient Data
10	Bill characteristic:	= YES	Group is false	Check next condition	Group is true

	Characteristic type = Tax Not Applicable indicator				
20	Characteristic Collection: Characteristic type = Tax Not Applicable	= YES	Group is false	Group is true	Group is true

NOTE:
The second criterion included above is used to evaluate the applicability of the tax on calculated adjustments. Since these adjustments are not yet linked to the bill, the first criterion cannot be used to evaluate the applicability of the tax threshold. The pre-bill completion algorithm C1-CPBC-TAXT instead adds the characteristic to the characteristic collection for use by the rate application when evaluating the adjustment's rate's eligibility rules. Refer to the ADJG-RT algorithm type for more information about the base package algorithm.

FASTPATH:
For more information, refer to Designing Calculation Groups and Rules.

An additional calculation rule is required to bill for the tax rounding discrepancy as follows

- Calculation Rule type = Service Quantity
- UOM/TOU/SQI = the identifier of the service quantity containing the tax discrepancy amount
- Value Type = Unit Rate
- Value Source = Value
- Value = 1

Other Financial Transaction Topics

Various topics about financial transactions are discussed in this section.

The Source Of GL Accounts On Financial Transactions

The following table lists the major financial events, their standard accounting, and the source of distribution codes used to derive the GL accounts sent to your general ledger.

Financial event	GL Accounting	Source Of Distribution Code
Create a normal utility bill segment. Bill Segment FT Algorithm is Payoff Amt = Bill Amt / Current Amt = Amt Due	Debit: A/R	SA Type

	Credit: Revenue / Taxes Payable	Calculation Rule
Create a bill for company usage. Bill Segment FT Algorithm is Payoff Amt = 0 / Current Amt = 0	Debit: Company Usage Expense	SA Type
	Credit: Revenue / Taxes Payable	Calculation Rule
Create a bill for charity. Bill Segment FT Algorithm is Payoff Amt=0 / Current Amt = Bill Amt	N/A - charity bills have no effect in the GL	N/A
	N/A	N/A
Create a payment segment for a normal utility service agreement	Debit: Cash	Bank Account on the Tender Source of the Tender Control for the Payment Segment's Tender.
	Credit: A/R	SA Type
Create a payment segment for a charitable contribution service agreement	Debit: Cash	Bank Account on the Tender Source of the Tender Control for the Payment Segment's Tender.
	Credit: Charity Payable	SA Type
Create a payment segment for auto-pay at bill completion time	Debit: Cash	Bank Account on the Tender Source on the Auto-pay Route Type of the Auto-pay Source.
	Credit: A/R	SA Type
Canceling a payment	Debit: A/R	SA Type
	Credit: Cash	Bank Account specified by the user on the cancel tender page. Note that this defaults to the original tender's bank account.
Create an adjustment to levy a charge	Debit: A/R	SA Type
	Credit: Revenue	Adjustment Type

The bottom line is as follows:

- If a bill segment has a financial effect, the distribution code to debit comes from the distribution code on the SA Type, the distribution code to credit comes from the calculation rule(s) used to calculate the bill segment.
- Payment segments always have a financial effect; the distribution code to debit comes from the bank account on the tender source of the tender control of the tender, the distribution code to credit comes from the SA type.
- If an adjustment has a financial effect, the distribution code to debit and credit comes from the SA type and adjustment type. If the adjustment is positive (i.e., the customer owes your organization more money), the distribution code to debit comes from the SA type; the distribution code to credit comes from the adjustment type. Vice versa if the adjustment is negative.

Defining Customer Options

The definition of a customer is someone (or something) with financial obligations with your company. These obligations ensue because the customer has agreed to purchase goods or services at an agreed price.

You may be surprised to learn that there is no "customer" record in the system. Rather, the system subdivides customer information into the following records:

- **Person.** The person record holds demographic information about your customers and every other individual or business with which your company has contact. For example, in addition to customers, person records also exist for landlords, contractors, accountants at corporate customers, guarantors of customers, energy distributors, collection agencies, etc.
- **Account.** Accounts are the entities for which bills are produced and therefore you must create at least one account for every person who has financial obligations with your company. The account record contains information that controls when the bills are created and how the bills are formatted.
- **Service Agreement.** Think of a service agreement as a contract between your company and the customer. The service agreement contains the terms and conditions controlling how the bill details are created. Every account will have at least one service agreement (otherwise, nothing will appear on the account's bills).

Before you can define persons, accounts, and service agreements, you must set up the control tables defined in this section.

●

FASTPATH:
For more information about how persons, accounts and premises are used by your customer service reps, refer to Understanding The "V".

➤

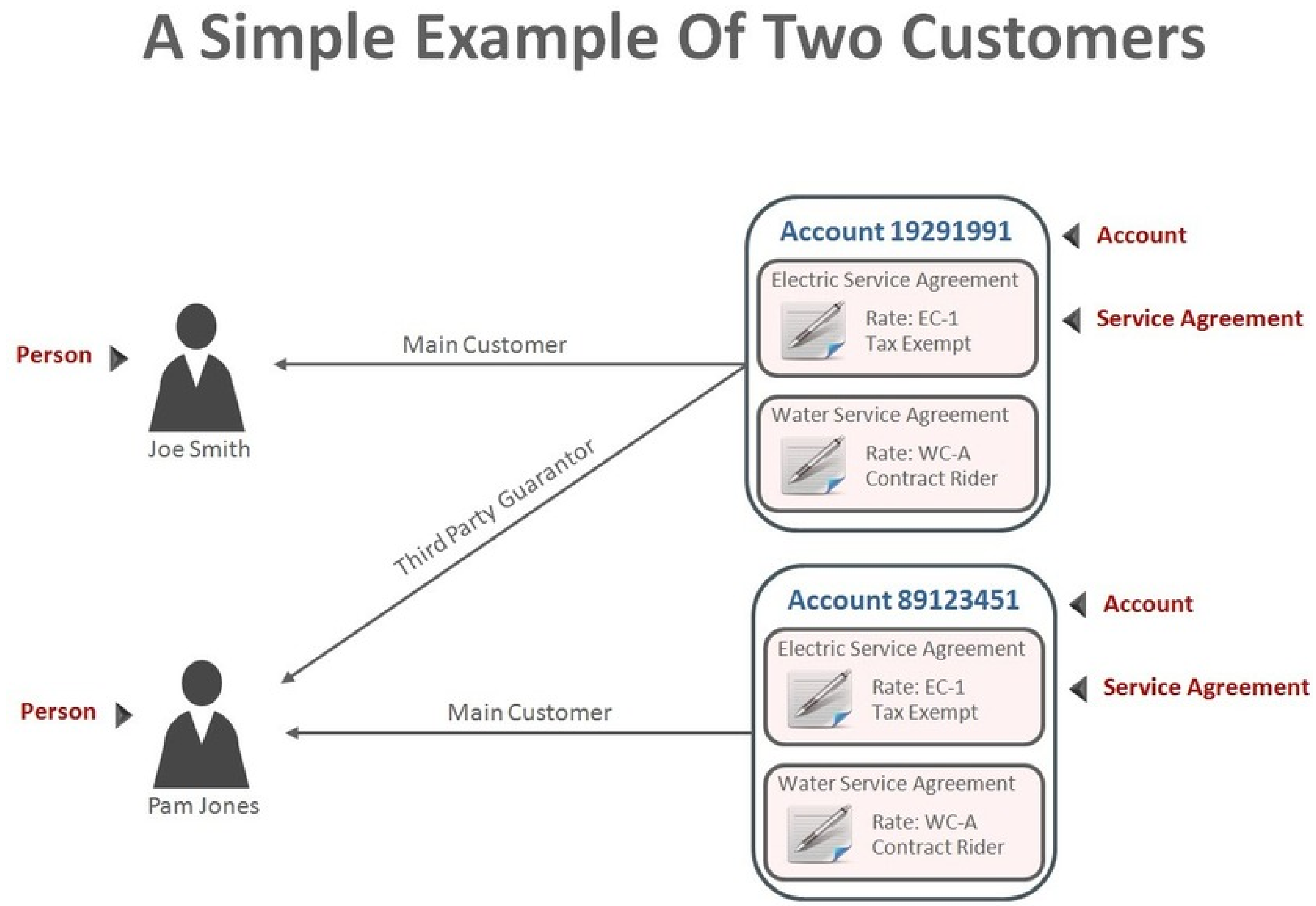
NOTE:
The tables in this section are only some of many tables that must be set up before you can bill your customers for the service(s) they consume. In this section, we limit the discussion to those tables that control basic demographic and financial information. In later sections, we describe the tables that control other billing-related functions like bill creation algorithms, meter reading and rates. It is only after all of these tables are set up that you will be able to generate bills and record payments.

Customer Overview

This section describes how the person, account, and service agreement records are used to record your customers' demographic and billing options.

A Simple Example Of Two Customers

The following picture illustrates two customers: Joe Smith and Pam Jones. Joe is the "main customer" on his account. Pam is the "main customer" on her account. Pam is also the "third party guarantor" on Joe's account.



Persons

Person records hold demographic information about the individuals and businesses with whom your organization communicates. Demographic information includes phone number(s), names and aliases, identification numbers, life support equipment needs, employment information, etc.

In the above example, 2 person records would be needed; one for Pam Jones and another for Joe Smith.

A new person is added when you first have contact with a person; the person does not have to be a customer before it is added. So, for example, if your company is starting a new marketing campaign, you can add information about potential customers the moment they are identified.

NOTE:
Businesses are persons too. In addition to humans, you use person records to maintain basic information about the businesses with which your organization has contact.

FASTPATH:

For a description of the control tables that must be set up before you can define a person, refer to Setting Up Person Options.

Accounts

An account is analogous to an account at a bank:

- A person or business with no financial dealings with a bank will have no account (but the bank may choose to keep demographic information about the person as part of their marketing efforts). The exact analogy exists in this system.
- Individuals with financial dealings with a bank will have one or more accounts. The number of accounts is up to the customer. The exact analogy exists in this system.

A simple way to determine the number of accounts a customer will have is to ask "how many bills do they want each period?" because a customer receives one bill for each account. For example:

- A residential customer who also owns a small business may choose to receive two bills each month; one for the residence, the other for the business. This way, the charges for their business would be segregated from their personal charges. This customer would have two accounts.
- A conglomerate that owns several factories may want their transportation gas charges to appear on a single bill rather than have a separate bill for each factory. This customer would have a single account.

Account ID Is Non-Intelligent

The unique number of an account is referred to as the "account ID". You are probably very comfortable with this concept. You may, however, have difficulty dealing with the fact that the account id in this system has no intelligence built into it (e.g., many systems include the bill cycle and geographic location in the account id). In this system, the account ID is a random, system-assigned value.

Because the account ID contains no meaning, it can remain with a customer for life, regardless of where they live, when they are billed, the type of service they receive, etc. This is important because it means that all of the financial history linked to the account remains with the customer for life.

NOTE:
The non-intelligence of the account ID is also important from the perspective of the parallel processing that takes place when the system creates bills. Because the collection of accounts to be billed in any given bill cycle will be randomly distributed through the number spectrum, the system can distribute account number ranges to parallel threads and each thread will process roughly the same number of accounts.

Account / Person Cross-Reference

A person may be linked to zero or more accounts. A person won't be linked to an account when they have no financial relationship with your organization. A person will be linked to multiple accounts when they have financial relationships with more than one account.

An account must reference at least one person (i.e., the main customer), but may reference an unlimited number of individuals. Multiple persons are linked to an account when several parties have some type of financial relationship with the account (e.g., third party guarantors, account contact, bill copy recipients, etc.).

When Is An Account Created?

A person can exist without an account until such time as the person formally requests the commencement of service. The moment the customer requests service, an account must be created (and the person must be linked to the account).

When Is An Account Expired?

Accounts never expire. Once a customer has an account, the account remains in the system forever. Linked to the account are service agreements that define the price and conditions of a service supplied to the customer. When an account has active service agreements, the system produces bills for it. If the account doesn't have active service agreements, the system will not produce a bill for it. You can think of an account without active service agreements as being "dormant", waiting for the day when the customer again starts service. If the customer never restarts, the account (along with its financial history) remains dormant forever.

Service Agreements

A service agreement is a contract (either formal or implied) between your organization and a customer. Every service agreement contains the price and conditions of a service supplied to a customer.

A service agreement is linked to an account. There is no limit to the number of service agreements that may be linked to an account.

When Is A Service Agreement Created?

A service agreement is created when the customer requests service (not when service commences). Typically, service agreements are created in the pending state and field activities are generated to connect service. When the field activities are complete, the service agreement becomes active and the billing process starts generating bill segments for the service agreement.

- **FASTPATH:**
  For more information about starting service, refer to The Big Picture Of Starting Service. For more information about bill segments, refer to Bill Details.

Financial Transactions Are Linked To Service Agreements

- **FASTPATH:**
  For more information about how financial transactions are linked to service agreements, refer to The Financial Big Picture.

When Is A Service Agreement Expired?

A service agreement is expired when the customer requests service be stopped. At that time, the service agreement is transitioned to the pending stop state and field activities are generated to stop service (these activities might involve simply reading the meter or they could involve disconnecting or removing the meter). When the field activities are complete, the system transitions the service agreement to the stopped state and the billing process generates a final bill for the service agreement. When the customer pays the final bill, the system transitions the service agreement to the closed state

● **FASTPATH:**
For more information about stopping service, refer to The Big Picture Of Stopping Service.

Setting Up Person Options

This section describes tables that must be set up before you can define persons.

Setting Up Identifier Types

When you set up a person, you may define the various types of identification associated with the person, e.g., their driver's license number, their tax identity, etc. Every piece of identification associated with a person has an identification type. These identifier type codes are defined using **Admin** > **Customer** > **Identifier Type.**

➤ **NOTE:**
How are person identifiers used? The reason why identifiers are defined on a person is so that users you can look for a customer using one of their person identifiers (see Control Central - Search Facilities for more information). In addition, person identifiers help prevent duplicate persons from being added to the database. This is because the system warns a user before they add a new person when a person exists with the same identifier.

Person identifier types are optional. An installation option controls whether at least one identifier type is required on every person.

Description of Page

Enter an easily recognizable **ID Type** and **Description** for the Identifier Type.

If the identifier type has a format against which validation can be performed, use **Identifier Format** to define the algorithm. To do this:

- Create a new algorithm (refer to Setting Up Algorithms).
- On this algorithm, reference an Algorithm Type that validates identifier types. Click here to see the algorithm types available for this plug-in spot.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference CI_ID_TYPE.

Setting Up Person Relationship Types

It is possible to associate persons to other person. For example,

- You might want to define the subsidiaries of a parent corporation
- You might want to define spouses as separate persons and then link each person to another person

When you link a person to another person, you must define in what way the person is related to the other person by using a person relationship type code. These codes are defined using **Admin** > **Person Relationship Type.**

Description of Page

Enter the following for each relationship type:

- Enter an easily recognizable **Relationship Type** code.
- Use **Description (Person1=>Person2)** to describe how the first person is related to the second person.
- Use **Description (Person2=>Person1)** to describe how the second person is related to the first person.

> **NOTE:**
> **Person1 versus Person 2.** When you link persons together, you do it in respect of one of the persons (which we call Person 1). For example, if you want to link the subsidiaries to a parent company, you do this in respect of the parent company (i.e., you define the parent company's subsidiaries using the Person - Persons transaction.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference CI_PER_REL_TYPE.

Setting Up Contact Routings

Base values for Contact Routing are supplied. You can add your specific contact routings using the **Contact Routing** extendable lookup. You can add more contact routings, but the ones supplied with the system cannot be edited or deleted.

> **NOTE:** Person contacts that reference the CSS routing type cannot be added by a user, and are not permitted to be edited or deleted. As well, records with this contact routing are only visible in the Person - Main page and are not displayed in the Account Portal or other places in the system.

Refer to Defining Extendable Lookups and for more information on lookups.

If your implementation is using notification preferences, you may need to set up delivery types before setting up contact routings. Refer to Setting Up Delivery Types for more information.

Setting Up Person Contact Status

Person contacts can have a status; for example, the status of an email contact could be pending, verified, or bounced. The product is delivered with statuses that are used in conjunction with opt-in functionality. When new person contacts are added, the initial status can be defaulted from the person contact type. See The Person Contact Status Can Be Controlled By A Process, *System Person Contact Statuses* and *Opt-in Process* sections for more information.

Your implementation may choose to introduce other logic to update the status. See The Person Contact Status Can Be Controlled By A Process for more information. You can add your specific person contact statuses by updating the value for the CND_VERIF_STATUS_FLG lookup field.

When new person contacts are added, the initial status can be defaulted from the person contact type. Refer to the Person Contact zone’s help text for more information.

Setting Up Person Contact Type Algorithms

Format Validation

Person contact information can be subject to format validation such as ensuring a phone number is entered in a particular format.

Click here to see the algorithm types available for this system event.

The base product provides the algorithms C1-VALEMFMT (Validate Email Format), C1-VALPHFMT (Validate Phone Format), and C1-VALANYFMT(Any Format Valid), which your implementation may use (and if so, no further configuration is needed). If your implementation's business rules require additional format validation algorithms, you may introduce your own.

NOTE: These algorithms confirm that a specific format exists and do not reformat or apply formatting to the corresponding field; for example, the *C1-VALEMFMT* (Validate Email Format) algorithm confirms that the email address contains required features (such as an @ symbol and domain). It will not, however, insert these features.

If your implementation's business rules require additional format validation algorithms, you may introduce your own.

Setting Up Person Contact Types

Person contact types define the format for entering person contacts and information about the person contact.

To set up contact types, select **Admin** > **Customer** > **Person Contact Type** > **Add.**

The Person Contact Type portal contains the following zones:

- Person Contact Type List
- Person Contact Type Zone

Person Contact Type Portal

This section describes the zones associated to the Person Contact Type Portal:

- Person Contact Type List
- Person Contact Type Zone

Person Contact Type List

The Person Contact List zone lists all person contact types. The following functions are available:

- Click a broadcast button to open other zones that contain more information about the adjacent person contact type.
- The standard actions of Edit, Delete and Duplicate are available for each person contact type.
- Click the Add link in the zone's title bar to add a new person contact type.

Person Contact Type Zone

The Person Contact Type zone contains display-only information about the Person Contact Type. This zone appears when a Person Contact Type has been broadcast form the Person Contact Type List zone or if this portal is opened via drill down from another page.

Please see the zone’s help text for information about this zone’s fields.

Where Used

Follow this link to open the data dictionary where you can view tables that reference C1_PER_CONTDET.

Choosing to Use Person Contact or Person Phone

Person contact is an updated and more configurable alternative to person phone and email. In this release, a feature is introduced that some system processes use to determine if person contact or person phone and email is being used. Person phone and email are referred to as legacy behavior. Legacy behavior is enabled when the Legacy Person Phone and Email **Option Type** on the Customer Information Options is set to Y. When set to N or not set legacy behavior is not enabled.

If your implementation uses functionality that has not been updated to use the feature, such as campaigns and order, you may wish to enable legacy behavior and continue to use person phone and email. It is important to note that Person Contacts are required for notifications.

The following system processes have been updated to alter functionality based on whether legacy behavior is enabled or not.

System Process	Behavior When Feature is Not Set	Behavior When Feature is Set
Person	The Person Phone Grid and the Person Email field are not shown	The Person Phone Grid and the Person Email field are shown. Person Contact is always shown.
Person	The functionality to maintain person contacts from person phone is not triggered even when a person contact type is associated with a phone type.	The functionality to maintain person contacts from person phone is enabled.
Account/ Person	When an email or fax bill or quote route type is chosen, then information displayed is related to person contacts.	When an email or fax bill or quote route type is chosen, then information displayed is related to person phone or email.
Start/Stop	The Person Contact collection is shown and can be edited.	The Person Phone collection and Person Email are shown and can be edited.
Case	The Preferred Contact Methods associated with legacy person phone and email are not shown.	The Preferred Contact Methods associated with legacy person phone and email are available. Person Contact is always available.
Customer Contact	The Preferred Contact Methods associated with legacy person phone and email are not shown.	The Preferred Contact Methods associated with legacy person phone and email are available. Person Contact is always available.

Person Contact Status Can Be Controlled By A Process

Overview

Person contact types can be configured to allow a status. For some person contact types, your implementation may want to control the status through a process instead of allowing the status to be set manually. An example of this is using person contact status to capture opting in for receiving text messages, which is required in some jurisdictions. Opt-in is seeking permission from a customer to use a person contact for a purpose such as sending a customer text messages.

Opt-in works in conjunction with the functionality to enable opt-in for a delivery type. Each can function independently, but together they offer a complete solution. Since they can function apart from each other, each is described independently. See Enabling Opt-in for a Delivery Type for more detail on this logic.

To control a person contact status with a process, the person contact type must first be configured to allow status. The person contact type must then be set up in the opt-in section of the **Notification Preferences**master configuration. By defining a Status Script, the person contact status can only be changed by the logic contained in or initiated from the script. The product is delivered with a script that creates an opt-in service task. This is designed specifically to support opting in to receiving notifications via text messages (SMS).

Opt-In Process

To use the delivered functionality, configure a new service task type for opt-in. This task type should reference the Notification Preference Opt-In Task Type Business Object (C1-NotifPrefOptInTaskType). The product is delivered with a service task business object The service task is responsible for initiating an opt-in request, updating the person contact status, and logging the user response. An inbound web service receives responses and transitions the service task. These components are designed to integrate with Notification Center. When a customer is sent an opt-in request, they are asked to confirm. The content of the message explains the commands to the user. Examples are CONFIRM or STOP. Notification Center is configured to interpret the specific commands and map them to those expected by the inbound service, which transitions the service task appropriately.

Aside from the initial opt-in request, users can stop or unsubscribe from notifications at any time. The delivered solution supports two modes for this functionality: unsubscribing from all notifications for a particular person contact or unsubscribing from a specific notification type for a particular person contact. This is controlled through configuration, and relies on the external system such as Notification Center to determine the notification type.

The **Notification Preferences**master configuration help text contains detailed information about the configuration required to control person contact status and setting up opt-in processes.

It is important to note that while the delivered process to control person contact status was designed to work in conjunction with the opt-in process, your implementation can design a custom process that can create a service task or do something completely different. Also, it can be independent of opt-in and notifications. For example, the process can be used to verify that an email address is valid by sending the customer a verification request.

There is another concept, which is enabling opt-in for a delivery type. This is closely related to the functionality described here, but has different implications in the system. See Enabling Opt-in for a Delivery Type for more information.

There is a limitation worth noting. A person contact type may be set up to allow multiple delivery types, however the delivered solution only allows a single delivery type to be associated with the process used to automate person contact status. Without this limitation, different processes would be responsible for updating a single person contact status and those updates could be conflicting.

The System Person Contact Statuses

The opt-in solution delivered with the product utilizes the three delivered status values. They are: **Pending**, **Approved**, and **Rejected**. Your implementation can add your own person contact statuses, but they will not work with the solution delivered with the product. See Setting Up Person Contact Status.

How the System Automates Person Contact Status

There are many ways that the process that controls person contact status can be invoked. For all of these, the person contact type must be set up in the **Notification Preferences**master configuration to have its status controlled by a process and, unless noted otherwise, the person contact is not already ***Approved***.

- The primary way to initiate the process to control person contact status is when a person contact is used to add a contact preference and the contact preference’s delivery type matches the delivery type associated with the person contact type as defined in the Notification Preferences master configuration. This method relies on enabling opt-in for a delivery type. This is how the tow processes work with each other. See Enabling Opt-in for a Delivery Type for more information. An example of this scenario is that a person contact exists and is of a type that can receive text messages, such as “Mobile Phone”. Until the person contact is used on a contact preference to receive a notification, there is no reason to initiate an opt-in process.
- This process can be automatically initiated by configuring a person contact to default to an initial status of ***Pending***. When a person contact is added, the process is initiated. An example of this scenario is an email person contact initiating a process to verify the email address as soon as it is added to the system.
- When the value of an approved person contact is changed, the process is reinitiated. For example, even though the approval is attached to a specific person contact in the system, which has a unique ID, externally the approval was for a particular phone number or email address. When the person contact value (i.e. the phone number or email address) is changed, the prior approval is no longer valid. The person contact status is changed to ***Pending***.

A user can manually initiate the process. A button is provided on the Person Contact row on Person - Main. The button can be used to reinitiate the process when the customer does not have the request and needs it resent. It can also be used to manually initiate the process upfront - before the person contact is used on a contact preference.

Maintaining Person Contact via Person Phone

Person phone is referenced extensively throughout the system. All system processes have been updated to determine if phone data and email data is being maintained on person contacts or being maintained on person phone and email. See Choosing to Use Person Contact or Person Phone for more information.

To keep person contacts records in sync with a person phone, you should consider which phone types should be stored as contacts, configure the person contact type corresponding to each phone type, and update the person contact type to reference the corresponding phone type. Many phone types may be associated with a single person contact type, but a single phone type may only be associated with one person contact type.

The base product provides a batch process C1-INPUS (Create Person Contact from Person Phone/Email) that creates a person contact for all person phone records whose phone type is associated with a person contact type. The batch process can also create person contact records for the email address on the person. You would run this process to create person contact records if you already have phone or email data for persons, and you wanted to start leveraging person contact as the repository for phone and email information. See the batch process description for more information.

> ➤ **NOTE:**
> There is a primary switch on person contact records; one person contact for each unique contact routing must be set as the primary contact. When the batch process creates person contacts from person phone data, the first person contact added for each unique contact routing is set as primary. This can be controlled to some extent by your implementation by associating phone types with person contact types sequentially and running the batch process multiple times. For example, you may prefer that a home phone be the primary over a cell phone if both exist. To ensure this happens, associate the home phone with a person contact type and run the batch process. Then associate the cell phone with a person contact type and run the batch process again. This will force all home phone records to be processed first. It is recommended that regardless of how many

times the batch processes run, they are run back to back. This should be thought of as a conversion of data and should be treated as such.

When a phone type is associated with a person contact type the following occurs:

- When a phone record is added, a corresponding person contact is added if one does not already exist.
- When a phone record is updated, corresponding updates are made to the person contact.
- When a phone record is deleted, the system tries to delete the corresponding person contact. All business rules for person contacts are enforced. If the person contact is associated with a contact preference, the person contact cannot be deleted, and in turn the phone record cannot be deleted.
- Person contacts associated with phone types cannot be deleted directly. The delete function is disabled.
- Contact information on person contacts associated with phone types cannot be updated directly. The contact information and extension are disabled.
- Person contacts whose person contact type is associated with a phone type cannot be added from the account contact information zone in the account portal. These need to be added through person phone.

There is no direct association between a person phone record and person contact (i.e. there are no foreign keys on either record pointing to the other). The system uses the association between phone type and person contact type and the specific phone number in order to find the associated person contact.

Your implementation may decide to use person contacts to store phone data and not associate them with phone types. Functionality related to the system use of person phone will not be available such as updating a phone number through an order.

If your implementation associates phone types with person contact types and wishes to remove this association, it is best to remove all phone types from all person contact types.

WARNING: Making changes to the association of phone type to person contact type can have serious implications for your data. Changing the person contact type associated to a phone type will likely result in difficulty maintaining both person phone and person contact data.

Setting Up Statement Construct Options

This section describes tables that must be set up before a statement construct can be set up for a person to begin receiving financial statements.

FASTPATH:
For more information, refer to The Big Picture of Complex Statements.

Setting Up Statement Route Types

Statement route types define the method used to route statements to persons. To define a statement route type, open **Admin** > **Billing** > **Statement Route Type.**

Description of Page

Enter a unique **Statement Route Type, Description** and **Statement Routing Method** for every statement route type.

NOTE:
The values for Statement Routing Method are customizable using the Lookup table. This field name is STM_RTG_METH_FLG.

The next two fields control how statements that are routed using this route type are printed (both in batch and online). Refer to Technical Implementation Of Batch Statement Production for more information about producing statements in batch. Refer to Technical Implementation Of Online Statement Production for more information about online statement production.

- Use **Batch Control** to define the process that creates the flat file that is passed to your statement printing software. If you use an **Extract Algorithm** to construct the downloaded information, you can use the STMDWLD process.
- Use **Extract Algorithm** to define the plug-in component that constructs the "flat file records" that contain the information to be merged onto statements routed using this route type. This algorithm is called when a user requests an online image of a statement on Statement - Main and it may also be called by the batch statement extraction process defined above. Click here to see the algorithm types available for this plug-in spot.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference CI_STM_RTE_TY.

Setting Up Account Options

This section describes tables that must be set up before an account can receive a bill.

Setting Up Account Management Groups

Users are informed that something requires their attention by entries that appear in To Do lists. For example, consider what happens when billing can't find a reading (and it's not allowed to estimate):

- The billing process creates a bill segment that is in error - meter read cannot be found.
- This error bill segment, in turn, triggers the creation of a To Do entry.
- The To Do entry is addressed to a role. A role is one or more users who can "action" the To Do entry.
- When a user views their To Do entries, they see all entries addressed to all roles of which they are part.

You can optionally use account management groups (AMG) to define the respective role to be assigned to To Do entries that are associated with an account and a given To Do type. For example, you can create an AMG called Credit Risks and assign this to accounts with suspect credit. Then, whenever an account-oriented To Do entry is created for such an account, it will be assigned a role based on the Credit Risks AMG. Refer to Assigning A To Do Role for more information..

NOTE:

Account management groups are optional. You need only set up account management groups (and link them to accounts) if you wish to address specific To Do entries associated with specific accounts to specific roles.

Account management groups are defined using **Admin** > **Customer** > **Account Management Group** > **Add.**

Description of Page

Enter an easily recognizable **Account Management Group** code and **Description** for each account management group. Use the grid to define the **To Do Role** to be assigned to entries of a given **To Do Type** that are associated with accounts that reference the **Account Management Group**.

➤ **NOTE:**
Only To Do entries that are account-oriented take advantage of the roles defined for an account management group (because only accounts reference an account management group).

Where Used

Follow this link to view the tables that reference CI_ACCT_MGMT_GR in the data dictionary schema viewer.

Setting Up Account Relationship Codes

When you link a person to an account, you must define in what way the person is related to the account by using an account relationship code. These codes are defined using **Admin** > **General** > **Account Relationship Type.**

Description of Page

Enter an easily recognizable **Relationship Type** and **Description** for each relationship type.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference CI_ACCT_REL_TYP.

Setting Up Alert Types

Account based alerts that appear in control central have an **AlertType**. To define valid alert types, navigate to **Admin** > **Customer** > **Alert Type.**

Description of Page

Enter an easily recognizable **Alert Type Code** and **Description** for each alert type. Specify the **Alert Days** to indicate the amount of time that alerts of this type will be effective by default. Specify a value of zero to indicate that alerts of this type will be effective indefinitely by default.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference CI_ALERT_TYPE.

Setting Up Bill Messages

There are various informational and warning messages that may appear on an account's bills. Each message is identified with a bill message code. To define a bill message code, open **Admin** > **Billing** > **Bill Message** > **Add.**

Description of Page

Enter a unique **Message Code** and **Description** for every bill message.

The following attributes control how and where the bill message appears on the customer's bill:

Priority controls the order in which the message appears when multiple messages appear on a bill.

> **NOTE:**
> The values for this field are customizable using the Lookup table. This field name is MSG_PRIORITY_FLG.

Insert Code controls whether a document should be inserted into the bill envelope when the bill message appears on a bill.

Message on Bill is the actual verbiage that appears on the customer's bill. If the message text is not static (e.g., field values need to be substituted into the body of the message), you can use the % *n* notation within the **Message on Bill** to cause field values to be substituted into a message. Refer to Substituting Field Values Into A Bill Message for more information.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference CI_BILL_MSG.

Setting Up Bill Route Types

Bill route types define the method used to route bills to accounts. To define a bill route type, open **Admin** > **Billing** > **Bill Route Type.**

Description of Page

Enter a unique **Bill Route Type** and **Description** for every bill route type.

Bill Routing Method controls the type of information that may be defined when the respective **Bill Route Type** is selected on Account - Person Information. The following options are available:

- Postal: Use this method if the routing is via the postal service.
- Fax: Use this method if the routing is via fax.
- Email: Use this method if the routing is via email.

> **NOTE:**
> The values for **Bill Routing Method** are customizable using the Lookup table. This field name is BILL_RTG_METH_FLG.

The next two fields control how bills that are routed using this method are printed (both in batch and online).

- Use **Batch Control** to define the background process that performs the actual download of the billing information. Refer to Technical Implementation of Printing Bills In Batch for more information about these processes.

- Use **Extract Algorithm** to define the algorithm that constructs the records that contain the information that appears on a printed bill. Refer to Printing Bills for more information.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference CI_BILL_RT_TYPE.

Setting Up Bill Cycles

●

FASTPATH:
Refer to Defining Bill and Measurement Cycles for a description of how to set up bill cycles.

Setting Up Customer Classes

When you set up an account, you must assign it a customer class. The topics in this section describe the customer class control table.

Customer Class - Main

To set up customer classes, navigate to **Admin** > **Customer** > **Customer Class** > **Add** and use the **Main page** to define your Customer Class.

Description of Page

Enter a unique **Customer Class** code and **Description** for every customer class.

Use **Collection Class** to define the collection class that defaults onto new accounts that belong to this customer class. An account's collection class may be subsequently modified if the account has special collection problems or needs.

●

FASTPATH:
For more information about the significance of collection class, refer to Designing Your Collection Classes.

Turn on **Business Activity Required** if service agreements linked to accounts with this customer class require a Business Activity description to be entered.

Turn on **Open Item Accounting** if accounts belonging to this customer class are subject to open-item account. Refer to Open Item Accounting for a complete explanation of the significance of this switch.

Turn on **Non CIS Payment** if accounts belonging to this customer class are used for payments made to reduce non-CIS debt. For example, assume your company accepts payments for a county assessor and you don't want to set up a separate account for each person who pays their assessment bill. You should set up the following information to accept such payments:

- Create a new customer class called "Non CIS Customer".
- Create a SA type for each type of non-CIS payment that customers can make. Make sure to enter a distribution code on each SA type that references the appropriate revenue (or payable) account. Don't forget to indicate that each SA type is not billed.

NOTE:
Payment Templates can be used for common types of non-CIS payment allocations. These templates are used to default the payment distribution and allow non-CIS payments to be directly allocated to specific distribution codes.

FASTPATH:
For more information about using Payment Templates to process non-CIS payments, refer to Non-CIS Payments.

- Create an account to which you'll book such payments. Have this account reference the new customer class. We recommend creating a separate account for each SA type that you created in the previous step.
- Create and activate a service agreement for the new account(s).

When someone pays for non-CIS debt, the operator will add a payment for the above account. On the payment, the operator should record reference information in order to know exactly why the payment was made. Refer to Payment Event - Main for more information.

You must define a variety of business rules for every division in which a customer class has customers. For example, if you operate in both California and Nevada AND you have CIS divisions for each state AND you have residential customers in each state, you must define **Customer Class Controls** for each CIS division. You do this on the Customer Class - Controls page. The grid that follows simply shows the CIS divisions for which business rules have been set up.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference CI_CUST_CL.

Customer Class - Bill Messages

When a customer class has bill messages, the system will sweep these messages onto bills created for accounts belonging to the customer class. Use this page to define a customer class's bill messages. Navigate to **Admin** > **Customer** > **Customer Class** > **Search** and then navigate to the **Bill Messages** page to maintain this information.

Description of Page

Use the bill messages collection to define **Bill Message** codes that should appear on bills that created for accounts that belong to a given customer class. For each message, also specify the **Start Date** and **End Date** when such a message should appear on the bill (leave **End Date** blank if the message should appear indefinitely).

Where Used

The system snaps customer class bill messages on a bill during bill completion. For more information about bill messages, refer to The Source Of Bill Messages.

Customer Class - Controls

You must define a variety of business rules for every division in which a customer class has customers. For example, if you operate in both California and Nevada AND you have CIS divisions for each state AND you have residential customers in each state, you must define **Customer Class Controls** for each CIS division in respect of the residential customer class. Open **Admin** > **Customer** > **Customer Class** > **Search** and then navigate to the **Controls** page to maintain this information.

Description of Page

The **Customer Class Controls** scroll contains business rules governing accounts that belong to a **CIS Division** and **Customer Class**. The following fields should be defined for each **CIS Division**:

- Use **Days Till Bill Due** to define the number of days after the bill date that the customer's bill is due. If the due date is a weekend or company holiday, the system will move the due date forward to the next workday (using the workday calendar defined on the account's CIS division).
- Specify the **Budget Plan** that defaults onto new accounts belonging to this customer class. Please note that an account's budget plan may be subsequently modified if the account has special budget processing needs. Refer to Setting Up Budget Plans for more information.
- Use **Min Credit Review Freq (Days)** to define the maximum number of days that can elapse between the reviews of an account's debt by the account debt monitor. Note, a value of zero (0) means that accounts in this customer class will be reviewed every day.
- Use **Credit Review Grace Days** to define the number of days after the bill due date that an account should be reviewed by the account debt monitor.
- Turn on the **Late Payment Charge** if customers in the class / division combination are eligible for late payment charges.
- Use **LPC Grace Days** to define the number of days after a bill's due date that a late payment charge will be generated (if the various LPC algorithms allow such - refer to How Late Payment Charges Get Calculated for the details). If the grace date falls on a weekend or holiday, the system moves the grace date to the next available workday (using the workday calendar defined on the account's CIS division).
- Specify an **Access Group** to default an access group onto an account based on the account’s CIS Division and customer class. This will override the access group defaulted from the user’s default access group. Note: the access group defined here will not apply if a Customer Class Control - Determine Access Group algorithm is defined and successfully returns an access group.

The grid that follows 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** 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.

> **WARNING:**
> These algorithms are typically significant system processes. The absence of an algorithm may prevent the system from operating correctly.

You can define algorithms for the following **System Events**:

System Event	Optional / Required	Description
Autopay Amount Over Limit	Optional	This algorithm is called to handle the situation when a system-initiated automatic payment is created that exceeds the customer's maximum withdrawal limit. Specifically, this algorithm is called when: - The account has a maximum withdrawal limit on their automatic payment options - The system attempts to create an automatic payment that exceeds this amount

		- The automatic payment algorithm that's plugged into the installation record has logic that invokes this algorithm when the above conditions are true If you do not plug-in this type of algorithm and the above situation is detected, the automatic payment will be created and no error will be issued. Refer to How To Implement Maximum Withdrawal Limits for more information. Click here to see the algorithm types available for this system event.
Bill Completion	Optional	When a bill for an account is completed, bill completion algorithms are called to do additional work. Refer to the description of the Complete button under Bill Lifecycle for a description of when this algorithm is called during the completion process. Click here to see the algorithm types available for this system event.
Bill Eligibility	Optional	Algorithms for this plug-in spot are called when generating a bill in batch billing. It provides the ability to determine if an account is ineligible for billing and should therefore be skipped from further processing. If an eligibility algorithm is not used, a bill is created for any account in the open bill cycle and is later deleted by the billing process if it detects that there is no information linked to the bill. Click here to see the algorithm types available for this system event.
Bill Segment Freeze / Cancel	Optional	When a bill segment for an account in this customer class / division is frozen or canceled, an algorithm of this type may be called to do additional work. Refer to Bill Segment Lifecycle for more information about freezing and canceling bill segments. Click here to see the algorithm types available for this system event.
Cancel Bill	Optional	This algorithm provides the ability to include additional cancel logic when canceling online. Algorithms of this type can be called in two modes: D (Determine Bill Page Buttons) and X (Cancel Bill). Mode 'D' governs whether an action button to cancel the bill will appear on the Bill page and mode 'X' performs the actual cancellation logic.

		Click here to see the algorithm types available for this system event.
Determine Access Group	Optional	When an account is added or an account's CIS Division or customer class is changed, this plug-in spot can be used to determine the appropriate access group. This plug-in spot allows more complex logic to be defined, such as checking the access group on the main person's others account's. If a determine access group is not used, the access group can be defaulted from the customer class control or the user. Click here to see the algorithm types available for this system event.
FT Freeze	Optional	When an FT is frozen, this algorithm is called to do additional work. For example, if you practice Open Item Accounting, you will need such an algorithm to handle the cancellation of match events when a financial transaction is canceled that appears on a match event. Refer to How Are Match Events Cancelled? for more information about cancellation. Click here to see the algorithm types available for this system event.
Late Payment Charge Eligibility	Required if the customer class / division is eligible for late payment charges	This algorithm is called by the late payment process to determine eligibility for late payments. Just because an account's customer class allows late payment charges to be calculated doesn't mean the account's delinquent service agreements will be levied late payment charges. In addition, a delinquent service agreement's SA type must reference a late payment charge algorithm. Refer to SA Type - Main for more information about SA type late payment charge issues. Refer to How Late Payment Charges Get Calculated for more information about late payment charges in general. ➤ **NOTE:** **Only One Algorithm**. Only one late payment charge eligibility algorithm may be defined for a customer class / CIS division combination.

		Click here to see the algorithm types available for this system event.
Levy an NSF Charge	Optional	This algorithm is called when a payment is canceled with a cancellation reason that indicates an NSF. Refer to NSF Cancellations for more information about what happens when a payment is canceled due to non-sufficient funds. **NOTE:** **Only One Algorithm**. Only one algorithm to levy an NSF charge may be defined for a customer class / CIS division combination. Click here to see the algorithm types available for this system event.
Order Completion	Optional	When an order is completed for a customer linked to this customer class, this algorithm is called to do additional work (e.g., create a customer contact). You need only specify this type of algorithm if you require additional work to be performed when an order is completed for customers who belong to this customer class. Click here to see the algorithm types available for this system event.
Overpayment Distribution	Required	When a customer pays more than they owe, this algorithm is called to determine what to do with the excess funds. Refer to Overpayment Segmentation for a description on how to configure the system to handle your overpayment requirements. **NOTE:** **Only One Algorithm**. Only one overpayment distribution algorithm may be defined for a customer class / CIS division combination. Click here to see the algorithm types available for this system event.

Override Due Date	Optional	An account's bill due date will be equal to the bill date plus its customer class' Days Till Due. If you need to override this method for accounts in a specific customer class, specify the appropriate algorithm here. ➤ **NOTE:** **Only One Algorithm**. Only one due date override algorithm may be defined for a customer class / CIS division combination. Click here to see the algorithm types available for this system event.
Payment Cancellation	Optional	Algorithms of this type are called when a payment is canceled. Click here to see the algorithm types available for this system event.
Payment Distribution	Required	This algorithm is called to distribute a payment amongst an account's service agreements. Refer to Payment Distribution for more information about how payment distribution works. ➤ **NOTE:** **Only One Algorithm**. Only one payment distribution algorithm may be defined for a customer class / CIS division combination. Click here to see the algorithm types available for this system event.
Payment Freeze	Optional	When a payment is frozen, this algorithm is called to do additional work. If you practice Open Item Accounting, you will need such an algorithm to link the payment's financial transactions to the match event that was originally created when the payment was distributed. Refer to Payments and Match Events for more information. Click here to see the algorithm types available for this system event.
Post Bill Completion	Optional	When a customer class has algorithms of this type, they are called after the completion of a bill for an account linked to this customer class.

		Refer to the description of the Complete button underBill Lifecycle for a description of when this algorithm is called during the completion process. Click here to see the algorithm types available for this system event.
Pre Bill Completion	Optional	When a customer class has algorithms of this type, they are called immediately before completion starts for an account linked to this customer class. These algorithms have the potential of: • Deleting a bill. You might want a pre completion algorithm to delete a bill if a condition is detected that should inhibit the sending of a bill to a customer (e.g., the bill just contains information about recent payments). • Aborting the completion process and creating a bill exception. If the algorithm indicates this should be done, the bill is left in the pending state and a bill exception is created describing why completion was aborted. You might want a pre completion algorithm to do this if, for example, integrity checks detect there is something wrong with the account or its service agreements. If the integrity check fails, the bill can be left in the pending state and a bill exception created describing why. Refer to the description of the Complete button under Bill Lifecycle for a description of when this algorithm is called during the completion process. Click here to see the algorithm types available for this system event.
Quote Completion	Optional	When aquote is completed for a customer linked to this customer class, this algorithm is called to do additional work (e.g., create a customer contact). You need only specify this type of algorithm if you require additional work to be performed when a quote is completed for customers who belong to this customer class. Click here to see the algorithm types available for this system event.
Write Off Method	Required if you allow users to write-off debt real time using the write off transaction	When a user presses the create button on the write off transaction, this algorithm is executed to write-off the selected debt. Refer to The Ramifications of Write Offs in the General Ledger for more information.

Click here to see the algorithm types available for this system event.

Setting Up Collection Classes

● **FASTPATH:**
Refer to Setting Up Collection Classes for a description of how to set up collection classes.

Setting Up Customer Information Options

When you add a new person, the system is set up by default to add an account for the person and go to the Start Service page when you save the new person information. You can change this functionality by configuring the following option types on the Customer Information Options feature configuration:

- **Add Account and Start Service Default:** Indicates whether the Add Account and Start Service option is selected on the person page, by default. If this option is not configured, the Add Account and Start Service option is selected by default.
- **Post Add Person BPA Script:** Indicates a BPA script to invoke when a user successfully adds or changes a person on the person page.

When you use control central to search for accounts, the system limits your search results based on your access rights. You can change this functionality by configuring the following option type.

- **Search All Accounts:** Indicates whether a Control Central search should allow a user to search all accounts without validating the user’s access rights. If the user tries to select an account without having the required access, they will not be able to navigate to the Account Information tab on Control Central for the selected account.

For more information about Feature Configurations, see Defining Feature Configurations.

Setting Up Customer Contact Options

This section describes tables that must be set up before you can define customer contacts.

● **FASTPATH:**
Refer to The Big Picture Of Customer Contacts for more information about customer contacts.

Setting Up Customer Contact Classes

Every customer contact record has a contact type that classifies the record for reporting purposes. And every contact type, in turn, references a customer contact "class". The class categorizes customer contacts into larger groupings for reporting purposes.

Open **Admin** > **Customer** > **Customer Contact Class** to define your customer contact classes.

Description of Page

Enter a unique **Contact Class** and **Description** for each customer contact class.

After you have created your customer contact classes, you'll be ready to setup your customer contact types.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference CI_CC_CL.

Setting Up Customer Contact Types

Every customer contact record has a contact type that controls the behavior of the customer contact.

- **FASTPATH:**
  Refer to The Big Picture Of Customer Contacts for more information about customer contacts.

Open **Admin** > **Customer** > **Customer Contact Type** > **Add** to define your customer contact types.

Description of Page

Every customer contact type is identified by a unique combination of **Contact Class** and **Contact Type**.

Enter a brief **Description** of the customer contact type.

Only specify a **Contact Shorthand** if customer contacts of this type can be added in the Customer Contact Zone. The value you specify in this field is what the user selects to add a customer contact in this zone.

If your implementation requires restricting customer contact types by CIS Division, specify a **CIS Division** on the customer contact type.

Use **Contact Action** if something should be triggered when customer contacts of this type are added. The only valid value in this release is Send Letter. If you select this option, you must also specify a **Letter Template**. Refer to Printing Letters for more information about how letters are produced.

Use the **Customer Contact Type Characteristics** collection to define characteristics that can be defined for contacts of a given type. Use **Sequence** to control the order in which characteristics are defaulted. Turn on the **Required** switch if the **Characteristic Type** must be defined on customer contacts of a given type. Turn on the **Default** switch to default the **Characteristic Type** when customer contacts of the given type are created. Enter a **Characteristic Value** to use as the default for a given **Characteristic Type** when the **Default** switch is turned on.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference CI_CC_TYPE.

Setting Up Notification Preference Options

The topics in this section provide background information about notification preferences.

Notification Preferences Overview

The Notification Preferences functionality is comprised of two main components: sending messages out of the system and maintaining customers’ communication preferences for receiving these messages.

Notifications are created by an algorithm plugged into the Create Notification plug-in spot on the Notification Type (BO). Click here to see the algorithm types available for this system event. The base product provides the C1-CRE-NTF algorithm, which creates the notification via an outbound message.

The following sections describe specific notification preference concepts and how to configure the system to use notification preferences.

Push vs. Subscription Notification Types

Notification types are defined as either push – which means that a customer need not sign up to receive notifications of this type, or subscription – which means that a customer must sign up to receive this type of notification. As you can tell, they have different uses and different behaviors.

Push notification types allow users to specify communication preferences for a general type of notification. These preferences include contact preferences and opt-out preferences, which are applied to individual notifications belonging to the general type. For example, "marketing" may be the general type, and the individual notification types may include several different types of marketing notifications that contain different messages and are targeted at different channels. Refer to Parent vs. Individual Push Notification Types for more information.

In the absence of any contact preference or opt-out preference, push-based notifications can determine a default contact and send the notification using it. The logic to determine a default contact is defined via an algorithm plugged into the Determine Default Contact plug-in spot on the Notification Type (BO). Click here to see the algorithm types available for this system event. The base product provides the C1-NTF-DFPRF (Determine Default Contact for Notification Type) algorithm, which looks to retrieve a default contact from the account's main person.

Subscription notification types define a particular notification and its message. Customers must have an active contact preference to receive these notifications. Subscription-based notification types can use an underlying service task. Refer to Some Notification Types Use Service Task for more information. A subscription-based notification type can also be set up to allow routing of bills or quotes to multiple destinations.

Parent vs. Individual Push Notification Types

Push notification types are configured as either a parent or individual notification type. An individual notification type references a parent notification type; a parent cannot reference another parent. Customers define communication preferences for parent push notification types. Notifications are sent out for individual push notification types. The preferences defined for parent push notification types are applied to the associated individual push notification types.

This means that customers do not have to supply their preferences for every individual notification that your implementation may send. Storing customer preferences for a parent notification type permits those preferences to be inherited for individual notification types added later in time under the same parent. The parent push notification types can be thought of as categories of notifications for which your customers define contact preferences and opt-out preferences.

When system processes or your custom processes attempt to send out a push notification, it should be for an individual notification type. The system will retrieve the parent notification type from the individual notification type when checking for communication preferences.

One function available to push-based notifications as described in Push vs. Subscription Notification Types is the ability to determine a default contact in the absence of a contact preference or opt-out preference. If your implementation wants to take advantage of this function, but would like the customer to supply preferences for receiving an individual notification, a parent push notification type can be set up with only one individual push notification type.

Delivery Type

Delivery Type is the channel used to send a notification. Examples are email, text message (SMS), fax, and outbound IVR. Delivery type is used throughout notification preferences. If your implementation is not using notification preferences, you would not utilize delivery type.

- Notification types list the supported delivery types. For example, a bill ready notification type may support email and SMS while a notification type to encourage customers to enroll in a budget plan may only support email. The message subject and body are configured for each notification type's supported delivery type.
- Person contact types list the allowable delivery types. For example, a cell phone person contact type may allow SMS while a home phone person contact type may not allow any delivery types. An email person contact type would allow an email delivery type unless the person contact type exists to capture a customer's email address and not to be used in conjunction with notification preferences. Note: the delivery types that can be specified on a person contact type are restricted to those defined on the person contact types contact routing. Refer to Setting Up Contact Routings for more information.
- A contact preference for a customer consists of the combination of notification type, delivery type, and person contact.

Enabling Opt-In for a Delivery Type

Overview

Opt-in can be enabled for a delivery type and is configured using the **Notification Preferences**master configuration; for example, opt-in can be enabled for SMS, but not email.

To help clarify some terminology, as explained in Person Contact Status Can Be Controlled by a Process, the product is delivered with a process that initiates a service task to request opt-in or permission to use a person contact. This solution requires that opt-in is enabled for the delivery type associated with the person contact type. In this regard, the two concepts are intertwined, but enabling opt-in for a delivery type can function independently.

Enabling opt-in is centered on the person contact status, while the process that controls the status can vary; therefore, it is possible to use enable opt-in for a delivery type while manually controlling the person contact status. Enabling opt-in for a delivery type has two key impacts system behavior.

- When a contact preference is added and the delivery type is enabled for opt-in, there is logic that checks if the person contact associated with the contact preference is in the ***Approved*** status. If it is not, and the person contact type is associated with a process to control the status, that process is initiated. If the person contact type is not associated with a process to control its status, a To Do Entry is created. The To Do Type is configured on the ***Notification Preferences*** master configuration. The system is delivered with To Do Type Opt-In Errors (C1-OPTIN) for this purpose.
- Notifications will not be delivered to person contacts that are not ***Approved***. When a system process initiates an outbound notification based on an active preference, if the person contact associated with the contact preference is not ***Approved***, a flag is set in the notification indicating that it should not be delivered. It is up to the receiving system, such as Notification Center, not to deliver the notification based on this flag. This is designed so that a record of the notification being generated is still captured. Also the C1-NTF-DFPRF (Determine Default Contact for Notification Type) will not choose a person contact that is not approved.

The **Notification Preferences**master configuration help text contains detailed information about the configuration required to enable opt-in for a delivery type.

Some Notification Types Use Service Task

Prior to the introduction of notification preferences functionality, the system supported notifications set up through self-service. These notifications used service tasks to store customer preferences and in some cases trigger notifications. The destination information for the notification existed outside of the system. Notification preferences, and more particularly communication preferences, define the user’s preferences on how to be notified for each notification type and where the notification is to be sent. This reduces, but does not eliminate, the need to use a service task to store a customer’s preference. These service tasks will not contain the self-service information when created internally by the system

There are four existing notification types that rely on service tasks to trigger a notification. Additionally, one of the prepaid notifications uses a service task type business object with specific fields for that notification type. Refer to Designing Notification Types for information regarding the configuration requirements for certain notification types.

Notification Preferences supports configuring notification types with or without a service task and only subscription-based notification types can be configured with an underlying service task.

Designing Notification Types

Some notification types require a specific configuration in order for the system to create notifications for that type.

Service Task Based Notification Types

There are system processes that rely on a hard-coded notification type lookup, referred to as notification type (legacy) on the notification type configuration. The system identifies the user-defined notification type from the lookup value.

The following identifies the configuration requirements for the notification types that are supported by existing processes in the system that rely on service tasks.

Notification Type (User Defined)	Push/Subscription	Controlled By	Notification Type (Legacy)	Service Task Type
Prepaid billing new charge	Subscription	Service Task	C1PC	Self-Service Task with BO C1-NotifyTaskType
Prepaid billing payment request	Subscription	Service Task	C1PP	Self-Service Task with BO C1-PPBPaymentNotifyTaskType
Payment Received	Subscription	Service Task	WSPR	Self-Service Task with BO C1-NotifyTaskType
Bill Ready	Subscription	Service Task	WXBR	Self-Service Task with BO C1-NotifyTaskType
Bill Due	Subscription	Service Task	WXBD	Self-Service Task with BO C1-NotifyTaskType
Late Payment	Subscription	Service Task	WSLP	Self-Service Task with BO C1-NotifyTaskType

Non-Service Task Based Notification Types

The following identifies the configuration requirements for the notification types that are supported by existing processes in the system that rely on the user-defined notification type to be configured.

NOTE:

> Marketing and forms notification types can be set up one of two ways: as subscription-based or as push-based. For it to be push-based, it would be an individual push, meaning that it references a parent push notification type specified by the implementer.

Notification Type (User Defined)	Push/Subscription	Controlled By	Notes
Marketing Preference (Lead Event)	Subscription or Individual Push	N/A	Notification type configured on lead event type of BO C1-LeadEvtTypeNotificationPref
Forms Update	Subscription or Individual Push	N/A	Notification type configured on form task type of C1-FormTaskTypeForNotification
Forms Closed	Subscription or Individual Push	N/A	Notification type configured on form task type of C1-FormTaskTypeForNotification

Bill and Quote Routing

Notification types can also be used to define contact preferences for bill and quote routing for email and fax. This can enable a bill or quote to create a routing for each active contact preference. For example for a single person on an account that receives a copy of the bill, the bill can be sent to more than one email address.

Some specific configuration is needed to support this.

The notification types are to be set up as follows:

Notification Type (User Defined)	Push/Subscription	Controlled By	Bill/Quote Route Type
Bill Route	Subscription	Bill Route	Specify the Bill Route Type
Quote Route	Subscription	Quote Route	Specify the Quote Route Type

Configure only the delivery type that corresponds with the bill or quote route’s routing method. For example, if the bill route type’s routing method is email, configure the email delivery type on the notification type.

Configure only person contact types whose contact routing corresponds with the bill or quote route’s routing method. For example, if the bill route type’s routing method is email, configure only person contact types with a contact routing of email. Ensure that the person contact type allows only the delivery type specified on the notification type.

This very specific configuration is needed because the system will override the existing logic that creates bill and quote routings when a notification type exists as described above and there are active contact preferences for that notification type with the specified delivery type and the person contact’s person contact type.

Setting Up Notification Type Algorithms

Suppression Criteria

Notification types can be suppressed from appearing in a list for a particular customer. This is done by plugging in a suppression criteria algorithm on the notification type.

Click here to see the algorithm types available for this system event.

The base product provides the C1-ALWY-SUP (Always Suppressed) algorithm, which suppresses the notification type for all customers. This does not impact active contact preferences for the notification type, nor does it prevent processes from creating notifications of this type. This should be used with business process changes to stop sending notifications of this type. This can also be used temporarily after a notification type has been configured, but before it should be used.

The base product provides the C1-CKSP-PPB (Check Prepaid Biller Suppression) algorithm, which suppresses the notification type if the customer does not have an active prepaid biller task. This should be used on prepaid related notification types so that only customers on prepaid billing receive these notifications.

If your implementation's business rules require additional notification type suppression rules, you may introduce your own algorithm.

Override Do Not Disturb

Do not disturb information can be captured on person contacts. Person contacts are associated with a notification type to define a contact preference. When notifications are sent out, do not disturb information is included in the message to the receiving system. Some notification types might be important enough to not honor the customer’s do not disturb information. For example, a notification might be sent to a subset of customers to warn of a gas leak and evacuation and needs to be sent regardless of the customers do not disturb wishes. This is done by plugging in an override do not disturb algorithm on the notification type. Note: this does not actually override the customer’s information, but sends an indicator in the message to the receiving system that the do not disturb information should be overridden or ignored.

Click here to see the algorithm types available for this system event.

The base product provides the C1-OVR-DND (Override Do Not Disturb) algorithm, which returns a value of true that is passed to the receiving system in the notification.

If your implementation has more specific business rules related to overriding a customer's do not disturb preference, you may introduce your own algorithm.

Setting Up Delivery Types

Base values for Delivery Type are supplied. You can add your specific delivery types by updating the value for the C1_SS_DELIVERY_TYPE lookup field.

Setting Up Notification Types

Each notification type defines the controls and behavior for defining communication preferences and sending out notifications of that type.

There are two main categories of notification types: subscription and push. These behave differently. Refer to Push vs. Subscription Notification Types for more information.

Push notification types are configured as either parent or individual push notification types. A parent push notification type is used to define communication preferences and an individual push notification type is used to send notifications. Refer to Parent vs. Individual Push Notification Types for more information.

The base product supplies one Notification Type business objects. For more details, refer to the specifications for this business object.

Some notification types require specific configuration. Refer to Designing Notification Types for more information.

To set up notification types, select **Admin** > **Customer** > **Notification Type +.**

The Notification Type Portal contains the following zones:

- Notification Type List
- Notification Type Zone

Notification Type Portal

This section describes the zones associated to the Notification Type Portal:

- Notification Type List
- Notification Type Zone

Notification Type List

The Notification Type List zone lists all notification types. The following functions are available:

Click a broadcast button to open other zones that contain more information about the adjacent notification type.

The standard actions of Edit, Delete, and Duplicate are available for each notification type.

Click the Add link in the zone's title bar to add a new notification type.

Notification Type Zone

The Notification Type zone contains display-only information about the Notification Type. This zone appears when a Notification Type has been broadcast form the Notification Type List zone or if this portal is opened via drill down from another page.

Please see the zone’s help text for information about this zone’s fields.

Where Used

Follow this link to open the data dictionary where you can view tables that reference C1_NTF_TYPE.

Setting Up the Outbound Message Type

An outbound message type is required for the notification outbound message. This outbound message type must reference the base C1-NotifPrefOutboundMessage (Notification Preference Outbound Message) business object.

Setting Up the Message Sender

A Message Sender is required to define how to send notifications. Use the context of the Message Sender to define the web service interface.

Setting Up the External System and Configure the Messages

Define an external system and configure the valid outbound message types and the method of communication for each (XAI, Batch, or Real Time); Real Time is generally the appropriate choice notifications).

Setting Up Notification Preferences Master Configuration

General configuration details for the notification preferences functionality is captured in the **Notification Preferences**master configuration.

For more information about specific fields in the master configuration, refer to the embedded help.

Other Edge Application Notification

Other applications such as *Oracle Utilities Network Management System* or *Oracle Utilities Meter Data Management* may initiate notifications; for example, the network management product can create notifications for outages or restoration information. The customer’s preferences to receive these notifications will reside in Oracle Utilities Customer Care and Billing, since this is where their contact information exists. The other applications need to be notified when an active contact preference is created for one of their notifications. The **Notification Preferences**master configuration provides the section to indicate if a Notification Type is owned by another edge application. When a contact preference is created or inactivated for one of these notification types, an outbound message is sent to notify the other edge product. When the system event in another edge application occurs that initiates a notification, that application calls and inbound web service that creates the notification.

The **Notification Preferences**master configuration help text contains detailed information about other edge applications.

Setting Up Service Agreement Options

This section describes tables that must be set up before you can define service agreements.

Setting Up Standard Industry Codes (SIC)

A service agreement for non-residential service should reference a standard industry code (SIC). This code is used to categorize service agreements for reporting purposes. To define a SIC, open **Admin** > **Customer** > **SIC Code.**

Description of Page

Enter a unique **SIC Code** and **Description** for the SIC.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference CI_SIC.

Setting Up Tax Exempt Types

Your rates will probably have provisions for the calculation of taxes of one type or another. Frequently you will have customers who are completely or partially exempt from these taxes. The service agreements for these customers will need to have tax exemption information in order for them to be billed properly. Tax Exempt Type is used to define the precise nature of the applicable exemption. To define the Tax Exempt Types you will use, open **Admin** > **Customer** > **Tax Exempt Type.**

Description of Page

Enter a unique **Tax Exempt Type** and **Description** for each type of tax exemption.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference CI_TAX_EX_TYPE.

Setting Up Contract Quantity Types

You may have customers whose contracts (service agreements) have contractual consumption limits. The service agreements for these customers must have information regarding this quantity in order to be billed properly. Contract Quantity Type is used to precisely define the nature of the quantity. To define the Contract Quantity Types, open **Admin** > **Rates** > **Contract Quantity Type.**

Description of Page

Enter a unique **Contract Quantity Type** and **Description** for each type of contract quantity.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference CI_CONT_QTY_TYP.

SA Type Controls Everything

Every service agreement references a SA type. The SA type controls all aspects of a service agreement's behavior including how service is started, how bills are created, how its financial transactions are booked in the general ledger, and much more. We don't explain how to set up SA types in this section because it's only after you have set up all of the control tables in this manual that you'll be able to finally define your SA types.

- **FASTPATH:**
  For more information about SA types, refer to Defining Service Agreement Types.

Financial Controls

- **FASTPATH:**
  There are also a number of control tables that must be set up to control the bills, payments, and adjustments that are linked to a service agreement. For more information about these tables, please refer to Defining Financial Transaction Options.

Setting Up Order Options

This section describes tables that must be set up before orders can be used to start service.

- **FASTPATH:**
  For more information, refer to The Big Picture of Campaigns, Packages and Orders.

Setting Up Column References

A column reference must be created for each miscellaneous field that's captured on an order that doesn't reside in a characteristic. Refer to Determine The Properties Of Every Miscellaneous Field for more information.

Open **Admin** > **Sales & Marketing** > **Column Reference** > **Add** to define your column references.

Description of Page

Enter an easily recognizable **Column Reference** code and **Description** for each column reference.

Specify the **FK Reference** to use if this column reference uses field values from another table. Use **Long Description** to describe the data that fields using this column reference capture.

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. You can define algorithms for the following system events: Post when order completed, Retrieve current value, Validate field value, Service task order processing, and Pre-process field value. Refer to Extract Column References for a description of these 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.

> **WARNING:**
> These algorithms are typically significant processes. The absence of an algorithm may prevent the system from operating correctly.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference CI_COL_REF.

Setting Up Order Cancellation Reasons

An order cancellation reason must be supplied when an order is cancelled. Open **Admin** > **Sales & Marketing** > **Order Cancel Reason** to define your reason codes.

Description of Page

Enter an easily recognizable **Order Cancel Reason** and **Description** for each order cancellation reason.

Where Used

Cancellation reasons are used when an order is canceled.

Setting Up Order Hold Reasons

An order hold reason must be supplied when an order is held. Open **Admin** > **Sales & Marketing** > **Order Hold Reason** to define your reason codes.

Description of Page

Enter an easily recognizable **Order Hold Reason** and **Description** for each order hold reason.

Where Used

Hold reasons are used when an order is held.

Setting Up Order Feature Configurations

Defining a feature configuration with a feature type of Order Configuration can increase performance of the Order page when campaigns have a large number of packages or criteria. Open **Admin** > **Feature Configuration** > **Add** to define a configuration for the feature type Order Configuration.

> **NOTE:**
> **Only one**. The system expects only one order configuration feature configuration to be defined.

Description of Page

The following points describe the various **Option Types** that may be defined:

- Eligibility Tree - Suppress Error Packages node. Select this option type and define a value if you would like the Order Eligibility Tree to suppress the node that contains packages with errors in their eligibility criteria. This is an optional setting. If the option type is not defined, the error packages node is displayed, if applicable.
- Eligibility Tree - Suppress Ineligible Packages node. Select this option type and define a value if you would like the Order Eligibility Tree to suppress the node that contains packages that are not applicable to the customer based on the eligibility criteria. This is an optional setting. If the option type is not defined, the ineligible packages node is displayed, if applicable.
- Eligibility Tree - Suppress Other Campaigns node. Select this option type and define a value if you would like the Order Eligibility Tree to suppress the node that contains other eligible campaigns. This is an optional setting. If the option type is not defined, the other campaigns node is displayed, if applicable.

Setting Up Program Management Options

This section describes how to set up the system for program management.

For more information about program management, see The Big Picture of Program Management

Designing Initiative Eligibility Criteria

The following describes guidelines for designing initiative eligibility criteria. These guidelines apply when using the base-supplied business objects and any additional business objects that the implementation defines.

There are two general types of initiative eligibility criteria business objects provided in base: specific and freeform. Your implementation decides whether to use only one of these types or a mixture of both.

> **NOTE:** Initiative criteria has three categories: eligibility criteria, related object criteria, and participation criteria. (The participation criteria type is currently not used.) During criteria maintenance and lead generation, the system identifies the category to use through a business object option called Initiative Criteria Type. Eligibility criteria business objects are defined with an Initiative Criteria Type of either C1EC (Eligibility Criteria) or C1BT (Both Eligibility Criteria and Participation Criteria).

Specific Criteria Business Objects

Most of the base-supplied business objects are designed to perform comparisons to a specific field. The algorithm type for retrieving/determining the value to compare with is already preset in the business object, such that the Algorithm values that the user sees on the user interface are restricted to the instances of that algorithm type.

Refer to the following base business objects for more details:

- Communication Channel Criteria
- Conservation Program Criteria
- Contract Option Criteria
- Rate Schedule Criteria
- SP Type Criteria
- Service Type Criteria
- Is Account Current Criteria
- Account is on Autopay Criteria
- Account is on Budget Criteria
- Number of Days Since Last Lead Criteria
- Account Customer Class Criteria
- Average Service Quantity Criteria
- Premise Cities Criteria

To add similar business objects:

- Identify the specific value for comparison. This could be a reference to an object or entity in the system, or a character string; such as, true/false.
- Develop the algorithm program to retrieve or determine the comparison value.
- Configure your algorithm type using the Initiative Criteria - Derive Field Value algorithm entity and reference the program you created.
- Configure your business object schema so that the Algorithm field retrieves all algorithm instances of the algorithm type you created.
- Define algorithm instances of your algorithm type.

Freeform Criteria Business Object

The Freeform Criteria business object, on the other hand, is designed to perform comparisons for different types of values. The comparison value is determined by a Derive Value Algorithm Type that is defined as an option on this business object. A number of these Derive Value Algorithm Types are supplied with this business object and your implementation can add to these, as necessary.

> **NOTE:** The supplied Derive Value Algorithm Types perform similar comparisons as those in the base-supplied specific criteria business objects (listed above). The Freeform Criteria business object should only be used if the available specific criteria business objects are not sufficient to perform the needed field value comparisons.

To add Derive Value Algorithm Types:

- Identify the specific value for comparison. It could be a reference to an object or entity in the system, a character string (such as, true/false), or a scalar value (such as, number of days or number of kwh).
- Develop the algorithm program to retrieve or determine the comparison value.
- Configure your algorithm type using the Initiative Criteria - Derive Field Value algorithm entity and reference the program you created.
- Define an algorithm instance for your algorithm type.
- Update the Freeform Criteria business object to add a Derive Value Algorithm Type business object option referencing your algorithm.

Repeat the above steps for each specific field to compare.

Designing Related Object Criteria

The following describes guidelines for designing related object criteria. These guidelines apply when using the base-supplied business objects and additional business objects that the implementation defines.

An initiative defines the related object(s) to associate with leads via the Object Nomination algorithm on the initiative business object. The lead generation process executes this object nomination algorithm after executing initiative eligibility criteria. The base supplied parent Initiative business object (C1–Initiative) is configured with an object nomination algorithm that retrieves the premises that are associated with the account’s active and pending start service agreement’s service points. This algorithm applies the related object criteria, which is configured on the initiative, to each premise. If the premise is determined to be eligible, it is returned as a related object for the lead. To override the base object nomination algorithm, add another instance of the Object Nomination System Event and specify your own algorithm.

The base product supplies related object criteria business objects that look for one or more premise to associate with a lead.

There are two general types of related object criteria business objects provided in base - specific and freeform. Your implementation decides whether to use only one of these types or a mixture of both.

> **NOTE:** Initiative criteria has three categories: eligibility criteria, related object criteria, and participation criteria. (The participation criteria type is currently not used.) During criteria maintenance and lead generation, the system identifies the category to use through a business object option called Initiative Criteria Type. Related object criteria business objects are defined with an Initiative Criteria Type of C1EO (Related Object Criteria).

Specific Criteria Business Objects

The base product supplies two business objects that perform comparisons to a specific premise-related field. The algorithm type for retrieving or determining the comparison value is preset in the business object, such that the Algorithm values that the user sees on the user interface are restricted to the instances of that algorithm type.

Refer to the following base business objects for more details:

- Premise SP Type Criteria
- Geographic Value Criteria

To add similar business objects:

- Identify the specific comparison value. It could be a reference to an object or entity in the system (such as, SP Type), or any other attribute that can be used to find the related object.
- Develop the algorithm program to retrieve or determine the comparison value.
- Configure your algorithm type using the Initiative Criteria - Derive Field Value algorithm entity and reference the program you created.

- Configure your business object schema so that the Algorithm field retrieves all algorithm instances of the algorithm type you created.
- Define algorithm instances of your algorithm type.

Freeform Premise Criteria Business Object

The Freeform Premise Criteria business object, on the other hand, is designed to perform comparisons to different types of values. The comparison value is determined by a Derive Value Algorithm Type that is defined as an option on this business object. Derive Value Algorithm Types are supplied with this business object and your implementation can add to these, as necessary.

➤ **NOTE:** The supplied Derive Value Algorithm Types perform similar comparisons as those in the base-supplied specific criteria business objects (listed above). The Freeform Premise Criteria business object should only be used if the available specific criteria business objects are not sufficient to perform the needed field value comparisons.

To add Derive Value Algorithm Types:

- Identify the specific comparison value.
- Develop the algorithm program to retrieve or determine the comparison value.
- Configure your algorithm type using the Initiative Criteria - Derive Field Value algorithm entity and reference the program you created.
- Define an algorithm instance for your algorithm type.
- Update the Freeform Premise Criteria business object to add a Derive Value Algorithm Type business object option referencing your algorithm.

Repeat the above steps for each specific field to compare.

Setting Up Lead Event Types

Each initiative can define a structured marketing effort for the events that take place during the life of a lead. For example, one event could be that a letter or an email is sent to the customer.

Each event references a lead event type. An initiative defines a lead event type profile that specifies one or more lead event types and the sequence in which those event types should occur. Refer to The Big Picture of Initiatives for more information.

Some types of lead events may also involve sending outbound messages to an external system — e.g., to notify a third party representative about a customer who has signed up. The lead event type business object can specify the outbound message type(s) to use.

Event types that involve additional workflow activities can define one or more steps; for instance, when work is outsourced to a third party representative (after customer sign-up), the third party lead event includes a list of steps to complete. Each event step has information (such as, completion date) that can be used to track progress.

The base product supplies a number of Lead Event Type business objects. For more details, refer to the specifications for these business objects.

To set up lead event types, select **Admin** > **Sales & Marketing** > **Lead Event Type.**

The Lead Event Type portal contains the following zones:

Lead Event Type List

The Lead Event Type List zone lists all lead event types. The following functions are available:

- Click a broadcast button to open other zones that contain more information about the adjacent lead event type.
- The standard actions of **Edit**, **Delete** and **Duplicate** are available for each lead event type.

Click the **Add** link in the zone's title bar to add a new lead event type.

Lead Event Type

The Lead Event Type zone contains display-only information about the Lead Event Type. This zone appears when a Lead Event Type has been broadcast form the Lead Event Type List zone or if this portal is opened via drill down from another page.

Please see the zone’s help text for information about this zone’s fields.

Where Used

Follow this link to open the data dictionary where you can view tables that reference C1_LEAD_EVT_TYPE.

Overriding Initiative Sign Up Options

The behavior of the Sign Up action on Lead maintenance is controlled by two business object options on the initiative:

- Sign Up UI Map
- Sign Up Service Script

When invoking a lead’s Sign Up action, the system retrieves these options and displays or executes them accordingly.

The base-owned parent initiative business object, C1–Initiative, is supplied with the Sign Up Service Script option referencing the C1-IntvSignU script. To override this, add another *Sign Up Service Script* option with a higher sequence number.

The Sign Up UI map, on the other hand, is not plugged in on the base BO, because this could vary across implementations. The Sign Up UI map C1-SignUpResponseCh provided in base simply prompts for a Response Channel. Implementations can use this map or their own specific map.

> ➤ **NOTE:** The system does not issue an error if the Sign Up UI map is not specified. It proceeds with executing the Sign Up Service Script and, if there are no problems, displays a sign up confirmation. The base-supplied C1-IntvSignU script includes logic to recheck the account’s eligibility for the initiative and update the *Has Signed Up* indicator on the lead.

Program Management Master Configuration

The master configuration business object for Program Management defines general configuration details for the program management functionality.

Navigate using **Admin** > **General** > **Master Configuration.** In the Master Configuration list, scroll to Program Management Configuration and click the add icon if there is no record yet or the edit icon to modify an existing record.

For more information about specific fields in the master configuration, refer to the embedded help.

Setting Up Letter Templates

You can set up a customer contact type to generate a form letter whenever a customer contact of this type is added. In fact, this is the only way to generate a letter in the system.

●

FASTPATH:
Refer to Printing Letters for more information about how letters are produced.

Every customer contact that causes a letter to be sent must reference a unique letter template. To define a letter template, open **Admin** > **Customer** > **Letter Template.**

➤

NOTE:
Document composition application users. If you use the document composition software to produce letters, there will be a template in the software associated with each letter. The name of the template must be the same as the code associated with the letter template set up in the system.

Description of Page

The following fields are required for each letter template:

- **Letter Template** is the unique identifier of the letter template.
- Use **Description** to enter a brief description of the letter.
- Turn on **Special Extract** if this type of letter should only be created via a system generated event such as a collection letter. Turning on this switch is what prevents a user from adding a customer contact that references this type of letter template (because you don't want a user to be able to request a letter associated with a system generate event by adding a customer contact, rather, they must execute the appropriate process and it will generate the customer contact).
- The next two fields control how letters of this type are printed (both in batch and online). Refer to Technical Implementation Of Batch Letter Production for more information about producing letters in batch. Refer to Technical Implementation Of Online Letter Production for more information about online letter production.
  - Use **Batch Control** to define the process that creates the flat file that is passed to your letter printing software. If you use an **Extract Algorithm** to construct the downloaded information, you can use the LTRPRT process.
  - Use **Extract Algorithm** to define the plug-in component that constructs the "flat file records" that contain the information to be merged onto letters of this type. This algorithm is called when a user requests an online image of a letter on Customer Contact - Main and it may also be called by the batch letter extraction process defined above. Click here to see the algorithm types available for this plug-in spot.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference CI_LETTER_TMPL.

Defining Service Order Options

Service orders manage the activities that are required to enable service, disable service, and perform other operations. These activities can range from simple events (such as, a meter read) to complex events (such as, installing a power line and a new meter). They can also include both fieldwork and smart commands.

Service Orders

This section provides an overview of service order activities.

Service Order Overview

Service Order Management provides functionality that orchestrates the field activities and smart meter commands that are necessary to change the state of service at a service point. This can involve complicated business logic due to numerous, often continually-changing variables, such as:

- The type of service order request; such as, enable service, disable service, cut, reconnect, and back-to back.
- Whether the service point is connected to the source of power, or not.
- If there is a device installed. If a device is installed, is it a smart meter, non-smart meter, or an item?
- The device's state at the service point; for example, pre-commissioned, connected, or disconnected.
- If it's a smart meters, what is the head-end system that manages the device at the service point/ If it's a non-smart meter (scalar), how close is the scheduled read date to the start date?

Additionally:

- Every implementation performs field work requests differently; for example, a cut service for non-payment is performed differently in every locale.
- Every head-end vendor requires different messages, in terms of the number and type of messages, and they change over releases of *their* software
- Field workers may perform different work than was requested by the system; for example, the system might request a simple reading, and the worker might decide to exchange the meter because the meter is old and broken..
- A service order request may be cancelled after it is scheduled or has been dispatched

Service orders are designed to resolve all of the above concerns, and provide a flexible and configurable solution to meet the challenge of managing customer service requests.

Understanding Service Order Management

At a high level, service order handles requests for service as follows:

Receive / Create Service Order Request

A service order request 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. Within Oracle Utilities Customer To Meter actions such as start service, stop service, and cut for non-payment result in the creation of a field activity request for each effected service point. These requests are sent to Service Order Management to be performed. In addition to requests that are created with Oracle Utilities Customer To Meter, service order requests can be received from other applications

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 the other activities needed to fulfill the original service request.

Evaluate Service Point and/or Meter

The service order activity 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

Based on the evaluation of the service point, meter, or item, the service order activity creates one or more activities perform the required actions.

- 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, or 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 Field Activity Request or Requesting System

When the orchestration activity determines that everything necessary to satisfy the service order request is ready, the orchestration activity is completed. If the orchestration activity was created by a field activity request, the field activity request is also completed. If, however, the orchestration activity was created due to external system rather than a field activity request the orchestration activity will inform the requester.

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, or item.

Supported Service Order Processes

Service Order Field Activity supports the following service order processes as part of the base package.

- Enable Service
- Disable Service
- Cut for Non-Payment
- Reconnect for Payment
- Meter Exchange
- Back-to-Back
- Cancelation Service Order Requests
- Update Service Order Requests

Enable Service

Enable Service is a request to start service for a particular service point. It is typically initiated by a *Start Service* field activity request, but can also be initiated by another CIS application.

A Start Service field activity request results in the creation of an Enable Service orchestration activity. The Enable Service orchestration activity then initiates the necessary field activities and smart meter commands to change the state of service at a service point to Enabled and confirm that an initial measurement exists; for example, an enable service orchestrating activity could involve the following service order field activities and smart meter commands based on the state of the service point:

- If there is no meter at the service point and the service point is not connected at the source, a service order field activity will be created to connect the service point to its source and install a meter.
- If a smart meter exists at the service point and it is in a disconnected state, a smart meter command will be sent to the appropriate head-end system to connect the meter.
- If there is a non-smart (manual) meter at the service point and it's On, a service order field activity will be created to request a read on or near the start date (depending when the next read is scheduled).

Disable Service

Disable Service is a request to stop service for a particular service point. It is typically initiated by a *Stop Service* field activity request, but can also be initiated by another CIS application.

A Stop Service field activity request will result in the creation of a Disable Service orchestration activity in Service Orders. The Disable Service orchestration activity will initiate the necessary field activities and smart meter commands to change the state of service at a service point to Disabled and confirm that a final measurement exists; for example, a disable service orchestration activity could invoke the following service order field activities and smart meter commands based on the state of the service point:

- If there is a non-smart (manual) meter at the service point and it's on, a service order field activity will be created to turn off the meter.
- If a smart meter exists at the service point and its state is connected or commissioned, a smart meter command will be sent to the appropriate head-end system to perform a remote disconnect of the meter.

Cut for Non-Payment

Disconnect for non-payment is a type of field work that is most often initiated from a collections process that has been created from within the Credit & Collection module. A collections process is created when the customer’s current balance has been determined to be past due. One of the methods for eliciting payment is to issue a disconnect. A *Disconnect for Non-Payment* field activity request will result in the creation of a Cut Service for Non-Payment orchestration activity in Service Orders. The Cut Service for Non-Payment orchestration activity will initiate the field activities and/or smart meter commands necessary to change the state of service at a service point to the disabled state.

For a Cut for Non-Payment service order, the orchestrations steps will include determining if a smart meter command can be executed, or if a request to a mobile field work application (such as Oracle Utilities Mobile Workforce Management) to dispatch a truck needs to be executed. Another example might be to determine if a disconnect can be executed for a service point. These rules might include determining if there is life support equipment at the premise, or if the service point can be disconnected at the current time of year or business hours based on utility defined rules taking into account predominant weather conditions (for instance, the utility cannot disconnect someone in the dead of winter or the service point resides at a postal code that is affected by extreme cold weather).

Reconnect for Payment

Reconnect Service for Payment is a request to restart service for a particular service point after customer has paid off their past due balance. It is typically initiated from the Credit & Collection's Severance Process.

A Reconnect for Payment field activity request will result in the creation of a Reconnect Service for Payment orchestrator activity in Service Orders. The Reconnect for Payment orchestration activity will initiate the field activities and/or smart meter commands necessary to change the state of service at a service point to the enabled state; for example, a service order request to reconnect service could involve the following service order field activities and smart meter commands based on the status of the service point:

- If there is a non-smart (manual) meter at the service point and it's turned off, a service order field activity will be created to turn on the meter.
- If a smart meter exists at the service point and it is in a disconnected state, a smart meter command will be sent to the appropriate head-end system to connect the meter.

Meter Exchange

Meter Exchange service is basically an exchange of an old meter with a new meter. A meter exchange request may be initiated as the result of:

- A smart meter rollout.
- A customer opting-out of having a smart meter.
- A meter no longer working correctly.
- A meter reaching the end of its life expectancy.
- A customer with a manual meter enrolling in a program that requires a smart meter.
- A need to upgrade measurement capabilities.

Meter exchanges can be initiated in a variety of ways, including:

- Within service order field activities (internally)
- By an external CIS or asset management system such as Oracle Utilities Operational Device Management
- A pick-up order from a field work application such as Oracle Utilities Mobile Workforce Management
-
-
-

Meter Exchanges involving smart meters will include smart meter commands in addition to field activities. A request can optionally include information on the type of meter needed, depending on the requesting system; this information could be specific (the type of configuration) or general (smart or manual).

Back-to-Back

Back to Back service is service disablement and enablement in single step; for example, one customer moving out while another moves in. The main purpose of this service is to minimize dispatching field work to obtain readings for both a start and stop event.

Another example of a back-to-back scenario would involve an tenant moving out and another moving into the same premise. As long as the gap between tenants (vacancy) is within the utility company's threshold, the change would be considered as a back-to-back event. In this situation, a single field task is used for this event to take the meter reading.

A Back-to-Back Service field activity request will result in the creation of a Back-to-Back Service orchestration activity in Service Orders. The Back-to-Back Service orchestration activity will initiate the field activities and smart meter commands necessary to change the state of the service of both customers at a service point to the desired state and confirm that measurements exists for that period.

Canceling Service Order Requests

A cancellation request basically cancels any orchestration and/or specific activities that are in progress. If the cancellation request is for an orchestration activity then that orchestration activity along with corresponding activities will be discarded. If the cancellation request is for a specific activity then only that specific activity will be discarded.

Depending on the status of a specific activity, a Cancel orchestration activity will either discard the activity or an outbound communication will be sent to the field work system to request a cancel the specific activity. Once the cancellation sent to the field work system is confirmed as successful, the Cancel orchestration activity will also cancel the parent orchestration activity and/or specific activities. If the cancellation sent to the field work system is not successful then the Cancel orchestration activity will be discarded. When it is discarded, it will send a response message to the requester indicating the cancellation was unsuccessful. A cancellation request can be initiated in a variety of ways, including by cancelling a field activity request , within Service Orders itself (internally), and by an asset management system such as Oracle Utilities Operational Device Management. Cancellation requests can also be created as pick-up orders from a field work application such as Oracle Utilities Mobile Workforce Management.

Update Service Order Requests

An update request handles requests for updates to an existing enablement, disablement, cut for nonpayment, and reconnect for payment orchestration activity and their related specific field activities.

The types of data available for update include the following:

- Instructions
- Comments
- Start Date/Time
- Contact Name
- Main Phone

Additionally, an update request can check to see if appointment information is different between the specific activity and the update orchestration activity. Depending on the status of the specific activity, direct updates will be made to the activities and communications within Service Orders or an outbound communication will be sent to the field work system to request an update of the specific activity fields that have been changed. Once that update to the field work system is confirmed as successful the Update orchestration activity will update the effected orchestration activity. If the updates are not successful the Update orchestration activity will be discarded. When it is discarded, it will send a response message to the requester indicating the update was unsuccessful.

Update service requests can be initiated by a field activity request or within Service Orders itself (internally).

Defining Disconnect Locations

When a service point is disconnected from the supply source, a disconnect location must be specified. This location defines where service was severed.

To define disconnect location codes, open **Disconnect Location** page.

Description of Page

Enter a **Disconnect Location** and **Description** for every disconnect location.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference CI_DISCON_LOC.

Defining Field Activity Profiles and Types

A field activity request initiates a service order to perform a task at a service point. Every field activity request references a field activity type.

Most field activity requests are created:

- When a customer service representative starts or stops service at a premise. When this happens, the system automatically creates field activity requests for each service point.
- When service is cut due to lack of payment.

If you set the system up correctly, your CSRs won't have to create these field activity requests. Rather, the system creates field activity requests based on the information you set up in your field activity type profiles.

- Refer to Starting Service & Field Activities for a description of how field activity requests are created to start service.
- Refer to Stopping Service & Field Activities for a description of how field activity requests are created to stop service.
- Field activities used to cut service due to non-payment are created by two modules:
  - If you use severance process to stop a service agreement due to nonpayment, refer to Field Events And Their Activities for a description of how these field activity requests are created.
  - If you use cut processes to stop a service agreement due to nonpayment, refer to Field Activities To Cut and Reconnect Service for a description of how these field activity requests are created.

The topics in this section describe how to design your field activity profiles and field activity types.

How Does A Field Activity Type Profile Get Used?

A field activity type profile contains:

- A list of the types of field activity requests that can be made for service points.
- A matrix defining the specific activity type(s) to generate in order to start service, stop service, or disconnect due to nonpayment.

You may wonder how field activity type profiles get related to your service points. It's a little indirect, but the indirection provides a great deal of flexibility:

- Every service point references an SP type.
- Every SP type references the specific field activity type profile used by the start/stop and credit and collections process to generate field activity requests.

An example will help illustrate how this works:

- When you start service at a specific service point, the system extracts the service point's SP type.
- Then, the system determines the field activity type profile that is to be used on this SP type.
- And finally, it generates the appropriate field activity request.

Designing Field Activity Type Profiles

With Service Order Management typically only a single field activity type profile will be needed for the meter and item related service points. This profile should have a template for each of the following with a service point condition of *External Fieldwork Orchestration* to indicate that Service Order Management is orchestrating the field work:

- Start Service
- Stop Service
- Start/Stop
- Cut for Non-Payment
- Reconnect for Payment

Designing Field Activity Types

A field activity type is required for the following events:

- Start service
- Stop service
- Cut for non-payment
- Reconnect for payment
- Back-to-back

Setting Up Field Activity Types

The topics in this section describe how to set up field activity types.

●

FASTPATH:
For more information about field activity types, refer to Defining Field Activity Profiles and Types.

Field Activity Type - Main

You begin to define a field activity type by opening **Admin** > **Field Work** > **Field Activity Type** > **Add.**

Description of Page

Enter an easily recognizable **Activity Type** and **Description** for each field activity type.

Use **Field Activity Priority** to define the priority associated with field activity requests of a given type.

> **NOTE:**
> The values for this field are customizable using the Lookup table. This field name is FA_PRIORITY_FLG.

Field Service Class is not applicable when Service Order Management is used.

Set **Fieldwork Orchestration** to Service Order Management

Eligible for Dispatch is not applicable when Service Order Management is used.

Appointment Booking is not applicable when Service Order Management is used.

Turn on **Display as Alert** if Control Central should display an alert if its premise has a completed field activity request of this type. If this switch is on,

- Use **Nbr Days Alert Active** to define the number of days the alert should appear on Control Central. The field activity request's scheduled date is used as the start date for the alert period.
- Enter the **Alert Information** to appear on Control Central.

> **NOTE:**
> **Recommendation.** We recommend only using this feature on unusual field activity types (e.g., cut for non-payment) so that a CSR is not presented with an alert for every field activity request.

Enter the **Business Object** for this FA Type if the FA type will need access to the BO plug-in spots. If the FA type has a business object specified, the system applies the business object’s rules when the FA is added, changed, or deleted. This includes the business object’s validation algorithms (only executed after the "core validation" specified on the maintenance object is done), post-processing algorithms, and audit algorithms. The maintenance object for the FA should specify an algorithm that finds its business object using its FA Type. Refer to the base package Determine Standard Business Object algorithm for an example of this type of algorithm.

Field Activity Step is not applicable when Service Order Management is used.

Use the **Characteristic** collection to define **Characteristic Type**s and their respective **Characteristic Values** to describe characteristics that are common to all field activity requests of this type.

> **NOTE:**
> You can only choose characteristic types defined as permissible on the field activity type record.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference CI_FA_TYPE.

Field Activity Type - FA Characteristics

Description of Page

Use the **Characteristics** collection to define characteristics that can be defined for field activity requests of a given type. Use **Sequence** to control the order in which characteristics are defaulted. Turn on the **Required** switch if the **Characteristic Type** must be defined on field activity requests of a given type. Turn on the **Default** switch to default the **Characteristic Type** when field activity requests of the given type are created. Enter a **Characteristic Value** to use as the default for a given **Characteristic Type** when the **Default** switch is turned on.

> **NOTE:**
> **Field activity requests created by the system**. When setting the Required switch, remember that most field activities are created by the system. Only turn on the required switch for these types of activities if a default characteristic value can also be indicated

Field Activity Type - FA Completion Control

Description of Page

> **NOTE:**
> **These algorithms are optional.** The use of completion algorithms on a field activity type is optional. You would only use them if you have special functions that should be executed when a given field activity type is completed.

The **Field Activity Completion Controls** tab is used when an algorithm should be executed when a field activity request is completed. The type of algorithm may differ based on the CIS division in which the service point's premise is located. The following fields must be defined:

CIS Division Defines the division associated for which the algorithm will be executed. The system will only execute the algorithm when a field activity request is performed at a service point whose premise is governed by the division.

FA Completion Algorithm Defines the algorithm that will be executed when a field activity request is performed at a service point whose premise is governed by the associated division. Click here to see the algorithm types available for this plug-in spot.

Field Activity Type - SP Type Review

Not applicable when Service Order Management is being used.

Setting Up Field Activity Type Profiles

The topics in this section describe the pages used to define a field activity type profile.

➤ **NOTE:**
When a new profile is added. When you introduce a new field activity type profile, you must define the SP types that use it. Refer to Understanding Service Point Types for more information.

● **FASTPATH:**
For more information about field activity type profiles, refer to Designing Your Field Activity Profiles & Types.

Field Activity Type Profile - Main

Description of Page

Enter a unique **Field Activity Type Profile** and **Description** for the activity type profile.

The **FA Type Profile Templates** indicate the templates that exist for this field activity type profile. Use the drill down button to go directly to the desired template. Alternatively, you can go to the **Template** tab and scroll until you find the correct template.

Field Activity Type Profile - Template

Description of Page

The information in the **Field Activity Profile Template** collection defines the field activity request(s) created for each situation identified by the **Customer Event**. The possible customer events are Cut for Non-payment, Disconnect Warning, Reconnect for Payment, Reread, Stop Service, Start Service, and Start/Stop.

➤ **NOTE:** Field activity requests do not need to be defined for Reread when Service Order Management is being used.

Other customer events can be defined on the Look Up page (search for the CUST_EVT_FLG field name). Refer to the following sections for more detail about customer events included with the base product.

The fields defined for each event are common. You define the field activity request(s) to be generated given the condition of the service point. The following fields display:

SP Field Condition Define the condition of the service point associated with the field activity. Define *External Fieldwork Orchestration* when Service Order Management is being used.

Sequence You will typically have a single field activity request for any specific combination of SP Field Condition and Disconnect Location. Therefore you'll just have a single sequence (say 10) for each combination. If you need to generate multiple field activities based on a given combination, use a unique sequence number for each activity.

No Activity Turn on this switch if no field activity should be generated for the condition.

Disconnect Location this is not applicable when Service Order Management is being used.

Activity Type Define the type of field activity request to be generated. You should take care to ensure that the activity type is defined as valid for the SP type (on the last page).

Where Used

Refer to the following sections for information about where each template is used.

Field Activity Requests Initiated To Start Service

Use the customer event Start to define the field activity request(s) used to start service at a service point whose SP type references this profile.

Description of Page

See Field Activity Type Profile - Template for a description of the fields.

Where Used

The start service process uses this information to determine the type of field activity requests to create to start service at a service point.

WARNING:
Warning. Field activity requests will only be created for starts if you have defined the appropriate field work creation algorithm on the service agreement's SA type. Refer to SA Type - Algorithms for more information.

Field Activity Requests Initiated To Stop Service

Use the customer event Stop to define the field activity request(s) used to stop service at a service point whose SP type uses this profile.

Description of Page

See Field Activity Type Profile - Template for a description of the fields.

Where Used

The stop service process uses this information to determine the type of field activity requests to create to stop service at a service point.

WARNING:
Warning. Field activity requests will only be created for stops if you have defined the appropriate field work creation algorithm on the service agreement's SA type. Refer to SA Type - Algorithms for more information.

Field Activity Requests Initiated For Back-to-Back Service

Use the customer event Start/Stop to define the field activity request(s) used to stop service for one customer and start service for another at a service point whose SP type uses this profile.

➤ **NOTE:**

> **Terminology.** We use the term **back-to-back** to describe the situation when a single field activity request supports both the stop and start service requests. The system sets up a back-to-back situation by default when it is aware of both the start and stop customers at a premise.

Description of Page

See Field Activity Type Profile - Template for a description of the fields.

Where Used

The start/stop service process uses this information to determine the type of field activity requests to create to start service at a service point.

Field Activity Requests Initiated To Cut Service Due To Non-Payment

Use the customer event Cut for Non-Payment to define the field activity request(s) used to cut service at a service point whose SP type uses this profile.

Description of Page

See Field Activity Type Profile - Template for a description of the fields.

Where Used

Severance and cut events use this information to determine the type of field activity requests to create to cut service at a service point.

Field Activity Requests Initiated To Place A Disconnect Warning At A Service Point

Use the customer event Disconnect Warning to define the field activity request(s) used to place a disconnect warning at a service point whose SP type uses this profile.

Description of Page

See Field Activity Type Profile - Template for a description of the fields.

Where Used

Severance and cut events use this information to determine the type of field activity requests to create to leave a disconnect warning at a service point.

Field Activity Requests Initiated To Reconnect Service At A Service Point

Use the customer event Reconnect for Payment to define the field activity request(s) used to reconnect service (after being cut) at a service point whose SP type uses this profile.

Description of Page

See Field Activity Type Profile - Template for a description of the fields.

Where Used

Severance and cut events use this information to determine the type of field activity requests to create to reconnect service at a service point.

Field Activity Requests Initiated To Reread A Meter At A Service Point

This template is not applicable when Service Order Management is being used.

Defining A Profile's Valid Field Activity Types

Description of Page

The **Field Activity Type** collection shows the field activity requests that may be created for at service points whose SP type references the field activity type profile.

Where Used

This information is used to control the types of field activity requests that may be created for at a service point.

Setting Up Fieldwork Cancellation Reasons

Description of Page

Enter a **Cancel Reason** and **Description** for every field activity request cancellation reason.

Only use **System Default** on those reason codes that are placed on field activity requests that are automatically canceled by the system. The following table lists the valid values and the condition where this cancel reason is used.

System Default	System Condition
Cut Process canceled	Placed on field activity requests that are canceled when a Cut Process is canceled.
SA Start/Stop canceled	Placed on field activity requests that are canceled when a pending start/stop is canceled.
Severance Process canceled	Placed on field activity requests that are canceled when a severance process is canceled.

NOTE:
Required values. You must have one reason code defined for each of the System Default values that corresponds to an event that may occur in your implementation.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference CI_FA_CAN_RSN.

Setting Up Field Activity Remarks

➤ **NOTE:** Field work remarks are typically processed by Service Order Management; therefore, field activity request remarks are not required.

Description of Page

Enter a unique **Field Activity Remark** and a **Description** for every field activity request remark.

Turn on **Eligible for Processing** if field activity requests marked with a given remark code should cause one or more algorithm to execute.

The grid contains **Algorithms** associated with the field activity request remark. 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.

The following table describes each **System Event**.

System Event	Optional / Required	Description
Field Activity Remark Activation	Optional	These algorithms are executed when there are pending FA remarks linked to a field activity request and the FACT (Field activity remark activation) background process runs. Refer to Field Activity - Characteristics/ Remarks for more information. Click here to see the algorithm types available for this system event.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference CI_FA_REM_CD.

Defining Outage Call Options

When you create an outage call, you must supply an outage call type. Outage call types contain a great deal of information that is defaulted onto the outage call, including the outage category group codes or trouble codes. To set up outage call types, select **Admin** > **Field Work** > **Outage Call Type** > **Add.**

➤ **NOTE:**

> Refer to The Big Picture of Outage Care Integration for more information about how trouble calls are created and sent to NMS.

The topics in this section describe the base-package zones that appear on the Outage Call Type portal.

Outage Call Type List

The Outage Call Type List zone lists every outage call type, i.e. every service task type that has a service task type class of Outage Call. The following functions are available:

- Click a broadcast button to open other zones that contain more information about the adjacent outage call type.
- Click the Add link in the zone's title bar to add a new outage call type.

Outage Call Type

The Outage Call Type zone contains display-only information about an outage call type. This zone appears when an outage call type has been broadcast from the Outage Call Type List zone or if this portal is opened via a drill down from another page. The following functions are available:

- **Edit**: to start a business process that updates the outage call type.
- **Delete**: start a business process that deletes the outage call type.
- **Duplicate**: to start a business process that duplicates the outage call type.

Please see the zone's help text for information about this zone's fields.

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 a separate installation of Oracle Utilities Customer Care and Billing)
- A field work system (such as Oracle Utilities Mobile Workforce Management)
- 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:

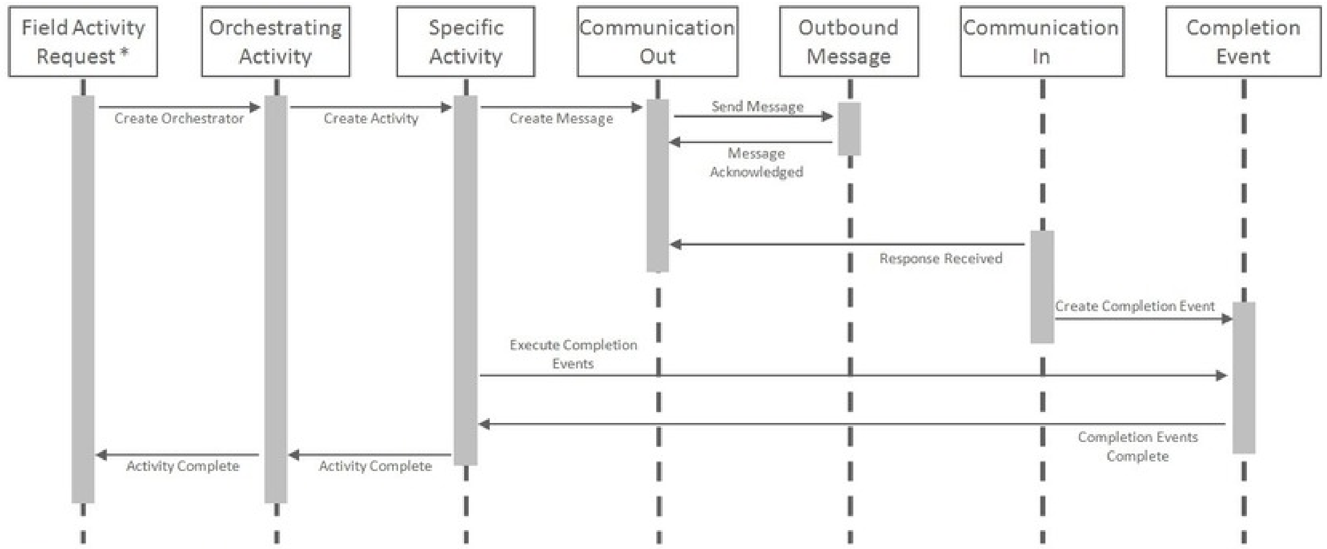
- **Activity Notification**: Used to send notifications to subscribing and/or requesting systems about the status of orchestration and/or service order field activities.
- **Appointment Request**: Used to send a request for an appointment to the field work system.
- **Cancelation Activity**: Used to send notifications to requesting systems when canceling orchestration and/or service order field activities.
- **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.

Setting Up Service Order Activities

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:



*** Field Activity Request:** Created by business processes; such as, start service, stop service, and credit & collection. A field activity request initiates an orchestrating activity to fulfill the request. *Note:* orchestrating activities can also be created manually and through external application requests.

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 an "Enable Service" could 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 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

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. An Enter algorithm sends an acknowledgement to the requesting system. The activity is transitioned to the next state via a monitor process.
Validate	Enter algorithms perform the following: Validate Activity Type Validate Service Point 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. Activities in this state can be corrected and retried.
Discarded	Activities discarded in other states enter this state. Enter algorithms perform the following: Validate that non-final child activities can be discarded without the need for a cancel activity Cancel non-final child activities 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. 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. 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. 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. 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 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. 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. Enter algorithms perform the following: Check to determine if there are child service order field activities in progress that have outbound communications awaiting a response. 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).

State	Description
Completed	Orchestration activities enter this state when all child activities have successfully completed. 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:

- 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.

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:

- **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 uses 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 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:

Mnemonic	Description
servicePointConnected	Specifies if the service point is currently connected Valid values are “true" and "false".
disconnectLocation	Specifies the “Disconnect Location” for the service point. Valid values are “D1SR" (source) and "D1DV" (device).
deviceInstalledAtSP	Specifies if there is a device currently installed at the service point. Valid values are “true" and "false".

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 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. Note: If this mnemonic is specified, the "Field Activity BO" parameter should specify the 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. 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. This allows the activity business object to be dynamically determined based on an algorithm instead of the "activityBOToCreate" or "specificFieldTask" mnemonics. 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
```

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. These parameters use 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. Note: If this mnemonic is specified, the "Field Activity BO" parameter should specify the 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. 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.
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. Note: If this mnemonic is specified, the "Field Activity BO" parameter should specify the activity business object to create. 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
```

Other parameters used by algorithms of this type include:

- **Field Activity BO**: Specifies the 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.
- **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.

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 Field Activity Types

Service order activity types must be configured for each type of service order activity.

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.

Configuring Service Order Activity Types

The Activity Type portal is used to display and maintain service order activity types.

Refer to Understanding Activity Types for more information.

You can access the portal from the **Admin** > **Communication** > **Activity Type**.

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

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.

Setting Up Service Order Field Activities

Understanding Service Order Field Activities

This section describes service order field activities and how they communicate with field work management systems such as Oracle Utilities Mobile Workforce Management.

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. Service Order Management supports integration with Oracle Utilities Mobile Workforce Management to support field activities, but can also integrate with other field work systems if needed.

Service Order 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?

At a high level, service order field activities work as follows:

Create a Service Order Field Activity Request

A service order field activity request 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. They can also be the result of other business processes; such as, a request to cut service due to non-payment. Within Oracle Utilities Customer To Meter actions (such as start service, stop service, and cut for non-payment) result in a field activity request being created for each affected service point. These requests are sent to Service Order Management to be performed. In addition to requests that are created with Oracle Utilities Customer To Meter, service order requests can be received from other applications.

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.

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. An Enter algorithm sends an acknowledgement to the requesting system. The activity is transitioned to the next state via a monitor process.
Validate	Enter algorithms perform the following: Validate Activity Type (and transition to error if invalid) Derive and validate service order recipient Validate duplicate and conflict service order activities Derive and validate service order service point Derive and validate service order device Validate address constituents Check for any existing cut service restrictions 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. Enter algorithms perform the following: Create a To Do based on specified To Do Type and To Do Role 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. 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. 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). 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.
Waiting for Appointment	If the service order field activity passes its validations and the effective date has been reached, the activity enters this state. Enter algorithms perform the following: Evaluate if an appointment is required for field tasks of this type. If not, the activity transitions to the "Communication in Progress" state. Create a To Do if an appointment is necessary but the system is not able to send an appointment request 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. Send a notification to the appointment handling system Monitor algorithms perform the following: Verify if an appointment has been supplied Send a notification to the appointment handling system

State	Description
	The activity is transitioned to the next state via a monitor process. 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. Enter algorithms perform the following: Create an outbound communication for the service order field activity (see Communication in Progress for more information) 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. Monitor algorithms perform the following: Check for existing child communications Check that the activity hasn't timed out
Discarded	Activities discarded in other states enter this state. Enter algorithms perform the following: Cancel outstanding outbound communications Cancel outstanding completion events Populate the cancel reason Send a failure notification to the requesting system Transition the parent activity to the " Activity Error" state (see Service Order Orchestration Activity Lifecycle for more information) Check if a Cancel Orchestration activity is required
Communication Error	If an outbound or inbound communication an Error state, the service order field activity enters this state. Monitor algorithms perform the following: Check that the activity hasn't timed out Enter algorithms perform the following: Create a To Do based on specified To Do Type and To Do Role 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. 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. Enter algorithms perform the following: Check to determine if there are associated outbound communications in progress. Cancel any outstanding outbound communications
Execute Completion Events	After an inbound communication is received, it enters this state. Enter algorithms perform the following: Executes completion events defined on the field task type (these completion events were initially created by the inbound communication).

State	Description
	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. The activity is transitioned to the next state via a monitor process. 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. Monitor algorithms perform the following: Check that the activity hasn’t timed out Enter algorithms perform the following: Create a To Do based on specified To Do Type and To Do Role 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. Activities in this state can be corrected and retried.
Completed	Service order field activities enter this state when all completion events have successfully completed. Enter algorithms perform the following: Update the parent orchestration activity Send a success response to the requester Transition the parent orchestration activity to the next state in its lifecycle Send a service order field activity completion outbound communication to subscribing systems.
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. Enter algorithms perform the following: Send a failed response to the requester Transition the parent orchestration activity to the "Activity Error" state. 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. The outbound message type is derived using the processing role defined on the "Send Notification to Appointment Handling System - Enter" algorithm.

Processing Role	Outbound Communication Business Object
Appointment Response	Send Appointment Response Outbound Message (D1-SendApptRespOutboundMsg)

Processing Role	Outbound Communication Business Object
(default) Used if: An appointment is required and needs to be scheduled Appointment has been set	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, such as Oracle Utilities Mobile Workforce Management. Communications can flow both outbound and inbound.

See Service Order Field Activity Communication for more information about service order field activity communication.

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	Field Activity Outbound Communication (D1-ActivityModifyOBComm)

Type of Outbound Communication	Outbound Communication Business Object
Used to send an update to a service order field activity previously sent to the field work system.	

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:

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 such as Oracle Utilities Mobile Workforce 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

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. 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 state, it creates an outbound communication.	Outbound Communication Business Object: Field Activity Outbound Communication (D1-FieldActivityOBComm)
3.	A Enter algorithm on the "Awaiting Response" state of the outbound communication retrieves	Enter Algorithm: Populate Send Detail for Field Activity (D1-POPSNDDTL)

Step	Process	Base Package Objects
	information needed by the outbound message to be sent to the field work system based on processing scripts specified on the field task type.	
4.	A Enter algorithm on the "Awaiting Response" state of the outbound communication creates an outbound message.	Enter Algorithm: Create Outbound Message (D1-COUTMSG) 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. Middleware components utilize Business Process Execution Language (BPEL).	External System: MWM Batch Control: Sync Request Monitor (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. The specific type of inbound communication business object created is determined by the Inbound Web Service.	Inbound Web Service: D1-FieldActivityIBComm Inbound Communication Business Object: Field Activity Inbound Communication (D1-FieldActivityIBComm)
9.	The inbound communication identifies the parent outbound communication.	Outbound Communication Business Object: Field Activity Outbound Communication (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. 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 D1-FieldActivityIBComm
11.	The inbound communication updates the outbound communication. 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 D1-FieldActivityIBComm Outbound Communication Business Object: Field Activity Outbound Communication (D1-FieldActivityOBComm)
12.	The outbound communication updates the “Completion Flag” and the original service order field activity business object.	Outbound Communication BO: Initiate Connect Disconnect (D3-InitiateConnectDisconnect) Field Activity Business Object: Field Activity (D1-FieldActivity)

Step	Process	Base Package Objects
	This update is performed by an Enter algorithm on the “Completed” Status of the outbound communication business object’s lifecycle.	

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, they 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 a set of criteria to allow the field crew to search for a service point 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.

Setting Up Service Order Field Activity Types

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.

Configuring Service Order Field Activity Types

The Activity Type portal is used to display and maintain service order field activity types.

Refer to Understanding Activity Types for more information.

You can access the portal from the **Admin** > **Communication** > **Activity Type**.

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

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.

Setting Up 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

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
	Connect SP at Meter: D1-ConnectSPAtMeter
	Connect SP at Source and Turn On: D1-ConnectSPAtSourceAndTurnOn
	Connect SP at Meter and Turn On: D1-ConnectSPAtMeterAndTurnOn
	Connect SP at Source and Install Meter: D1-ConnSPAtSrceAndInstMtr
	Connect SP at Meter and Install Meter: D1-ConnSPAtMtrAndInstMtr
	Install Meter: D1-InstallMeter
	Turn On Meter: D1-TurnOnMeter
Enable Service — Items	Connect SP at Source: D1-ConnectSPAtSource
	Item - Connect SP at Device: D1-ConnectSPAtDevice
	Item - Connect SP at Source and Turn On: D1-ConnSPAtSrceAndTurnOnDvc
	Item - Connect SP at Device and Turn On: D1-ConnectSPAtDvcAndTurnOn
	Connect SP at Source and Install Device: D1-ConnSPAtSrceAndInstDvc

Service Order Activity	Field Task Types
	Connect SP at Device and Install Device: D1-ConnSPAtDvcAndInstDvc Item - Install Device: D1-InstallDevice Turn On Item: D1-TurnOnItem
Disable Service — Meters	Turn Off Meter: D1-TurnOffMeter
Disable Service — Items	Turn Off Item: D1-TurnOffItem
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

Defining Credit & Collections Options

NOTE:
The functionality described in this section is meant to handle the collection of unpaid balances. If your organization practices open-item accounting and collects on unpaid bills, you will not use this functionality. Rather, you will use the functionality described under Defining Overdue Processing Options.

The system periodically monitors how much your customers owe to ensure they haven't violated your collection criteria. When a violation is detected, the system generates the appropriate responses (e.g., letters, disconnect notices, collection agency referrals, and eventually write off). This section describes how to set up the tables that control your credit & collections processing.

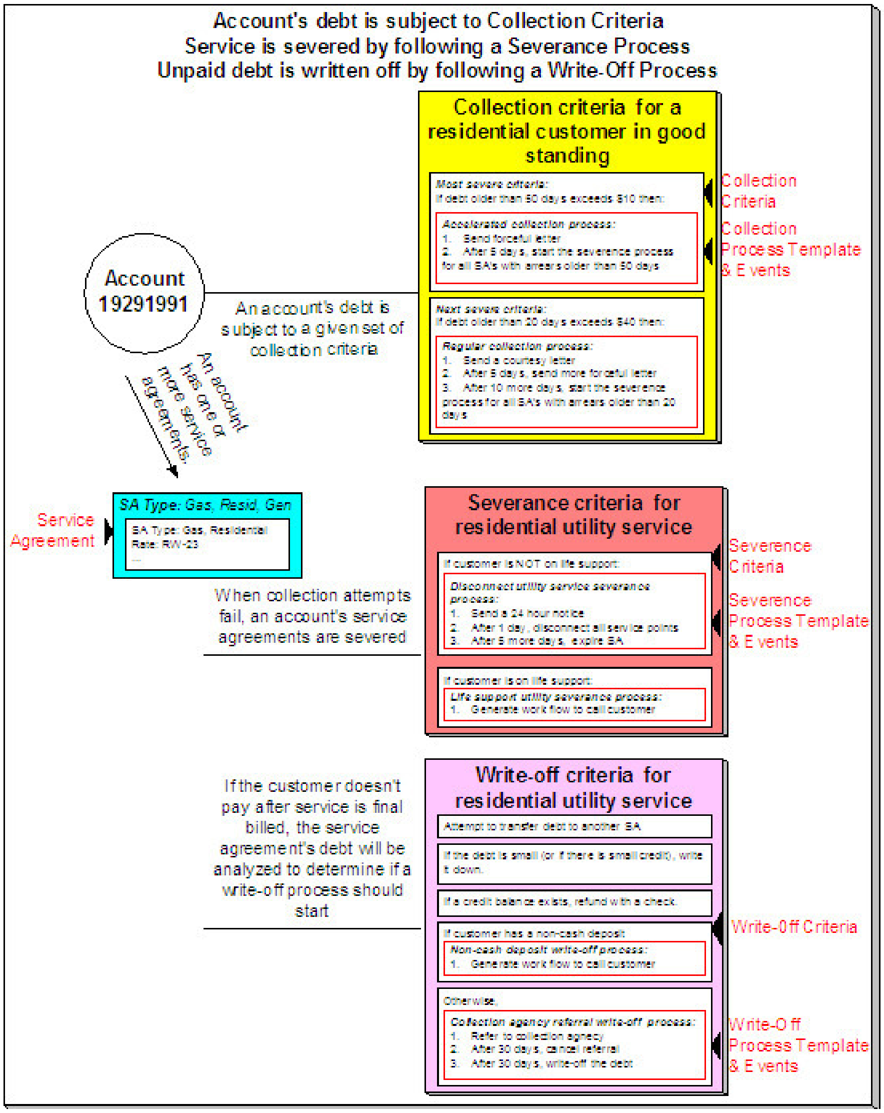
WARNING:
Setting up the credit & collections control tables is as challenging as your organization's collection rules. If you have simple rules then your setup process will be straightforward. If your collection rules are complicated (e.g., they differ based on the type of customer, the type of debt, the age of debt, the type of service, etc.), then your setup process will be more challenging.

The Big Picture Of Credit & Collections

This section provides an overview of important C&C concepts with which you should be familiar before you set up your C&C control tables.

Collection Criteria vs. Severance Criteria vs. Write Off Criteria

The following diagram introduces important concepts related to the C&C processes:


Account's debt is subject to Collection Criteria
Service is severed by following a Severance Process
Unpaid debt is written off by following a Write-Off Process
Account 19291991
An account's debt is subject to a given set of collection criteria
An account has one or more service agreements.
Collection criteria for a residential customer in good standing
Most severe criteria:
If debt older than 50 days exceeds $10 then:
Accelerated collection process:
1. Send forceful letter
2. After 5 days, start the severence process for all SA's with arrears older than 50 days
Next severe criteria:
If debt older than 20 days exceeds $40 then:
Regular collection process:
1. Send a courtesy letter
2. After 5 days, send more forceful letter
3. After 10 more days, start the severence process for all SA's with arrears older than 20 days
Collection Criteria
Collection Process Template & Events
Service Agreement
SA Type: Gas, Resid, Gen
SA Type: Gas, Residential
Rate: RW-23
...
When collection attempts fail, an account's service agreements are severed
Severance criteria for residential utility service
If customer is NOT on life support:
Disconnect utility service severance process:
1. Send a 24 hour notice
2. After 1 day, disconnect all service points
3. After 5 more days, expire SA
If customer is on life support:
Life support utility severance process:
1. Generate work flow to call customer
Severence Criteria
Severence Process Template & Events
If the customer doesn't pay after service is final billed, the service agreement's debt will be analyzed to determine if a write-off process should start
Write-off criteria for residential utility service
Attempt to transfer debt to another SA
If the debt is small (or if there is small credit), write it down.
If a credit balance exists, refund with a check.
If customer has a non-cash deposit:
Non-cash deposit write-off process:
1. Generate work flow to call customer
Otherwise,
Collection agency referral write-off process:
1. Refer to collection agnecy
2. After 30 days, cancel referral
3. After 30 days, write-off the debt
Write-Off Criteria
Write-Off Process Template & Events

There are many important concepts illustrated above:

An account's debt comes from its service agreements	An account's debt is managed at the service agreement level, i.e., the system keeps track of how much a customer owes in respect of each service agreement. In order to determine an account's balance, the system must add up the debt on each of the account's service agreements.
Collection criteria define intolerable debt	Collection criteria are control data that define intolerable debt. Most criteria are defined using a combination of number of days in arrears and a dollar amount.
Collection criteria may be compared to an account's total debt or to subsets of debt	If your organization has simple collection procedures, you will probably target collection criteria at an account's total debt. However, you have the option of segregating an account's debt into debt classes and targeting the collection criteria at each class. For more information about debt classes, see Different Collection Criteria For Different Customers And Different Debt.

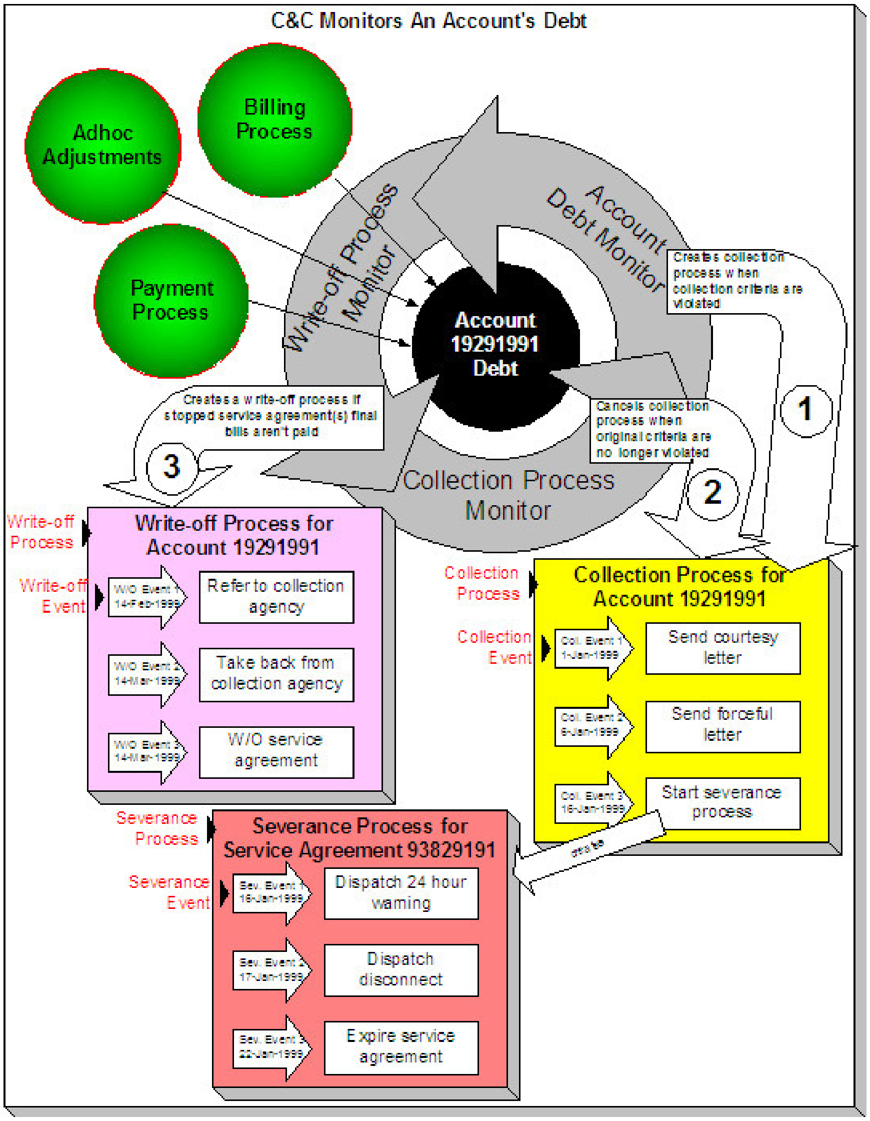
Collection criteria also define what to do when the level of intolerable debt is exceeded	When you define collection criteria, you also define how the system should respond if an account violates your criteria. These collection events are defined in respect of a "collection process template".
There are usually several collection events that take place when criteria are violated	A collection process template usually has several collection events. Each event is meant to prod the customer to pay. The initial collection events are typically letters. If payment is not received after several such attempts, the last collection event typically starts a severance process for each service agreement in arrears.
A severance process template defines how to sever a service agreement	A "severance process template" defines how to sever a given type of service agreement. A severance process template usually contains several severance events. These events are a series of letters and / or disconnection field activities that eventually result in the expiration of a service agreement if payment is not received.
Severance criteria define how to sever service agreements	Severance criteria define the severance process to be executed for service agreements of a given SA type. The severance process may differ depending on some attribute of the customer (or premise). For example, you may have a different severance process if the customer has life support equipment.
After a service agreement is severed, it will be final billed	When the last active service agreement linked to an account is stopped, the system changes the account's bill cycle to bill that evening. If only one of many SAs is stopped, the SA will only be final billed as per the account's original bill cycle schedule.
If a customer doesn't pay their final bill, the account's debt will be analyzed to determine if the system can reduce the debt to zero using a variety of mechanisms	The system will look at an account's finaled debt on its next scheduled credit review date (typically a few days after the bill's due date). The system will attempt to reduce the service agreement's debt to zero using all of the following methods: • If the account has active service agreements, it will transfer the finaled debt to an active service agreement. • If the debt or credit amount on the service agreement is small, the system will generate an adjustment to 'write it down' (or up in the case of a small credit). • If the service agreement has a large credit amount, the system will generate an A/P adjustment (resulting in a check being sent to the customer).
If a customer's finaled debt cannot be reduced via any of the previous methods, the system creates a write-off process	A write-off process contains one or more write-off events. These events can generate a letter, send a To Do entry to a CSR, send a referral to a collection agency, etc. When you set up the system, you define the type of write-off process to use for every collection class / write-off debt class combination. In addition, you can also indicate when the type of write-off process should differ depending on some attribute of the customer (or premise). For example, you may have a different write-off process if the customer has a non-cash deposit.
The last write-off event typically causes the debt to be written off	Ultimately, if the write-off events fail, the debt will have to be written off. When debt is written-off, the system creates a write-off service agreement and transfers the outstanding debt to it. This means the debt stays with the account for life and will have to be paid off if the customer ever returns.

NOTE:

Checkpoint. At this point, you should be familiar with the concept that an account's debt is compared to user-defined collection criteria. If the account violates the criteria, a series of events will ensue that prod the customer to pay. If the customer doesn't respond, every service agreement in arrears will be severed (i.e., disconnected). If lack of service doesn't inspire payment, the service agreement will be expired and a write-off process will be created to manage the write-off activities.

The C&C Monitors

Your collection, severance and write-off criteria described in the previous section exist to support the processes that manage the collection activities. The following diagram illustrates, at a high level, the major processes that manage the collection of overdue debt:



There are many important concepts illustrated above:

Bills, payments and adjustments affect an account's debt	An account's debt is the accumulation of all bills, payments and adjustments.
The Account Debt Monitor creates a collection process when an account violates collection criteria	Periodically, a background process referred to as the Account Debt Monitor (ADM, and ADM2 or C1-ADM) determines if an account's debt violates your collection criteria. If so, a collection process is created using the violated criteria's collection process template. Refer to When Is An Account's Debt Monitored? for a description of when an account's debt is compared against collection criteria.
A collection process contains one or more collection events	The collection process contains a series of collection events. These events correspond with the collection event types associated with the collection process template. The initial collection events are typically letters. If payment is not received after several such communications, the last collection event typically starts a severance process for each service agreement in arrears.
The Collection Process Monitor cancels a collection process when warranted	The Collection Process Monitor cancels a collection process when its service agreements satisfy your cancellation criteria (e.g., when the service agreements have less than $10 of debt older than 20 days). Refer to How Are Collection Processes Cancelled for more information about the cancellation process.
The last collection event starts one or more severance processes	The last collection event typically starts one or more severance processes. A severance process contains the activities necessary to sever a service agreement. The service agreement(s) that are severed may be all SAs that are associated with the collection process. Alternatively, you can nominate a service agreement to act as the primary service to cut (you'd do this if you cut electricity when the customer doesn't pay for their gas). The algorithm on the collection event that starts severance will control which service agreement(s) are severed. Refer to How To Nominate A Single Service Agreement To Sever for more information.
Each service agreement has its own severance process	Every service agreement that is severed has a severance process. The type of process is dependent on the severance criteria linked to the service agreement's SA type.
A severance process contains one or more severance events	The severance process contains a series of severance events. The events correspond with the severance process template's severance events.
The system cancels a severance process when warranted	The system cancels a severance process when its service agreement satisfies your cancellation criteria (note, it is possible to set up the system so that all service agreements in the debt class must satisfy your cancellation criteria before a severance process is cancelled). It's important to note that the cancellation is real time (as opposed to the cancellation of collection processes, which happens in a background process). Refer to How Are Severance Events Canceled? for more information.
The last severance event should expire the service agreement	The last severance event typically expires its service agreement. When the last service agreement linked to an account is expired, the system will schedule the account for billing (outside of its normal bill cycle schedule).
If you nominate a single SA to sever when multiple SAs are in arrears...	Earlier we indicated that you can nominate a service agreement to act as the primary service to cut (you'd do this if you cut electricity when the customer doesn't pay for their gas). If you do this, you also need

	a severance event that will sever all other service agreements in the debt class if the severance of the nominated service agreement doesn't inspire payment. A severance event algorithm to do such is supplied with the base package. Refer to How To Nominate A Single Service Agreement To Sever for more information.
The Write-Off Monitor creates a write off process to collect stopped, unpaid debt	The Write-Off Monitor reviews stopped and reactivated service agreements after their closing bill's due date (plus grace period). The Write-Off Monitor attempts to reduce the service agreement's debt to zero using all of the following methods: • If the account has active service agreements, it will transfer the finaled debt to an active service agreement. • If the debt or credit amount on the service agreement is small, the system will generate an adjustment to 'write it down' (or up in the case of a small credit). • If the service agreement has a large credit amount, the system will generate an A/P adjustment (resulting in a check being sent to the customer). If the system is unsuccessful in reducing the account's debt to zero, a write-off process will be created using the appropriate write-off process template. Refer to The Big Picture Of Write Off Processing for more information about the write-off process.
A write-off process contains one or more write-off events	The write-off process contains a series of write-off events. These events correspond with the write-off event types associated with the write-off process template. The initial write-off events are typically collection agency referrals and/or letters. If payment is not received as a result of such efforts, the last write-off event typically writes off the customer's debt.
The system cancels a write-off process when warranted	The system cancels a write-off process when its service agreements no longer have debt (i.e., they become closed).
Another write-off process will be created if a closed service agreement ever reactivates	If a service agreement becomes reactivated (e.g., because the final payment bounces), the service agreement will be processed by the Write-Off Monitor and the whole write-off process starts again.

NOTE:

Checkpoint. At this point, you should be familiar with the concept that a collection process will be created for an account that violates collection criteria. The collection process consists of a series of events that typically generate letters and / or To Do entries. If the customer doesn't respond, a severance process will be started for one or more service agreements. A severance process consists of a series of events that typically generate letters and/or disconnection field activities. If lack of service doesn't inspire payment, the last severance event expires the service agreement (and a final bill will be scheduled when the last service agreement is expired). If the customer doesn't pay the final bill, a write-off process will be created for each type of unpaid debt. The write-off process consists of a series of events that ultimately result in the write-off of the customer's debt. When debt is written-off, the system creates a write-off service agreement and transfers the outstanding debt to it. This means the debt stays with the account for life (because the write-off service agreement is linked to the account) and will have to be paid off if the customer ever returns.

The Big Picture Of Collection Processes

The topics in this section describe how collection processes are created and cancelled.

> **FASTPATH:**
> For more information refer to The Lifecycle Of A Collection Process And Its Events.

How Does The Account Debt Monitor Work?

This section describes how the Account Debt Monitor uses your collection criteria and collection process templates to collect overdue debt.

Different Collection Criteria For Different Customers And Different Debt

Consider the following:

- You probably have different collection criteria for different jurisdictions (i.e., CIS Divisions). For example, if you have customers in different states / provinces, you may have different regulator-imposed criteria applied to each state's debt. You differentiate your debt in respect of the collection process via the **CIS division code on each customer's account**.
- You probably have different collection criteria for different customer segments. For example, customers with large bills probably have strict criteria, whereas you're probably more lenient with small customers (or vice versa). You differentiate your customers in respect of the collection process via a **collection class code on the customers' accounts**. An account's initial collection class is defaulted from its customer class. You may override an account's collection class at will.
- You probably have different collection criteria for different classes of debt. For example, if a single customer has both regulated and unregulated debt, you probably have commission-imposed criteria to be applied to the regulated debt, but you have the freedom to apply stricter criteria to the unregulated debt. You differentiate your debt in respect of the collection process via a **debt class code on the customers' service agreements** (note: the debt class is actually defined on the service agreement's SA type).
- You will have different criteria for every currency in which you work because the monitoring process always compares a customer's debt against some value and this value must be denominated in the customer's currency. A customer's currency is defined using a **currency code on the account**.

Given the above, you should understand that different collection criteria will exist for every combination of CIS division, collection class, debt class, and currency code. If you're confused, consider the following matrix (where we assume you have a single currency and division and therefore avoid the third and fourth dimensions):

SA's Debt Class	Account's Collection Class:	Account's Collection Class:
	Commercial Customer	**Residential Customer**
Regulated	N/A - there is no regulated, commercial customer debt.	Highest Priority: If > $5 in arrears by more than 50 days, create the accelerated collection process for residential customers.

		Lower Priority: If > $25 in arrears by more than 25 days, create the courtesy reminder collection process for residential customers.
Unregulated	Highest Priority: If > $10 in arrears by more than 50 days, create the accelerated collection process for commercial customers. Lower Priority: If > $1000 in arrears by more than 25 days, create the normal collection process for commercial customers.	Highest Priority: If > $10 in arrears by more than 25 days, create the normal collection process for residential customers.
Charitable Contribution	Highest Priority: If > $10 in arrears by more than 50 days, create the charitable collection process.	Highest Priority: If > $10 in arrears by more than 50 days, create the charitable collection process.

Also, notice that there can be multiple criteria for each cell in the matrix. What differentiates one collection criteria from another is its priority. The higher priority criteria will be compared first. If the debt meets the criteria, the collection process is initiated and no further comparisons are performed.

FASTPATH:
For more information about maintaining this matrix, refer to Setting Up Collection Class Controls. For more information about how the system handles an element in this matrix that has multiple criteria, see How Is An Account's Debt Monitored?.

Override Conditions

WARNING:
Your credit & collection requirements may not require any overrides and therefore this section may not be relevant for your organization.

The matrix presented in the previous section showed:

- You can have different collection criteria for different categories of debt and customers.
- When a collection criteria is violated, the system generates a specific collection process.

This works great for many organizations, but if your organization has other factors that affect either the collection criteria OR the collection process that is initiated when the criteria is violated, you may need to use override collection criteria. For example,

- If you have a different collection process for regulated, residential debt during the winter months, you'll need to use override collection criteria (where the override criteria is "if it's winter").
- If you have different collection criteria for customers with a poor credit score, you'll need to use override collection criteria (where the override criteria is "if the customer's credit rating is poor").

FASTPATH:
Refer to Designing Your Collection Class Control Overrides for more information.

This section describes how and when the Account Debt Monitor analyzes an account's debt.

When Is An Account's Debt Monitored?

The account debt monitor (ADM) analyzes an account's debt at least every X days, where X is defined on the customer class control associated with the account's customer class and division (in the field Min Credit Review Freq (Days)).

In addition, an account's debt will also be monitored as follows:

- The ADM looks at an account's debt X days after an account's bill due date (X is defined on the account's customer class in the field Collection Grace Days).
- The ADM looks at an account's debt after a payment is canceled when the cancellation reason indicates NSF (non-sufficient funds).
- The ADM looks at an account's debt after a payment arrangement is broken (assuming you use the base package's break payment arrangement plug-in). Refer to Monitoring Payment Arrangements for more information.
- The ADM looks at an account's debt after a pay plan is broken. Refer to The Pay Plan Monitor for more information.

How Is An Account's Debt Monitored?

Assume the following collection control matrix exists for your organization:

SA's Debt Class	Account's Collection Class:	Account's Collection Class:
	Commercial Customer	**Residential Customer**
Regulated	N/A - there is no regulated, large customer debt	Highest Priority: If > $5 in arrears by more than 50 days, create the accelerated collection process for residential customers. Lower Priority: If > $25 in arrears by more than 25 days, create the courtesy reminder collection process for residential customers.
Unregulated	Highest Priority: If > $10 in arrears by more than 50 days, create the accelerated collection process for commercial customers. Lower Priority: If > $1000 in arrears by more than 25 days, create the normal collection process for commercial customers.	Highest Priority: If > $10 in arrears by more than 25 days, create the normal collection process for residential customers.

This matrix contains the information used by the Account Debt Monitor.

FASTPATH:
For more information about the information in this matrix, refer to Different Collection Criteria For Different Customers And Different Debt.

This matrix can be overwhelming when viewed as a whole. So let's consider how to use it for a specific account's debt and things will become clearer.

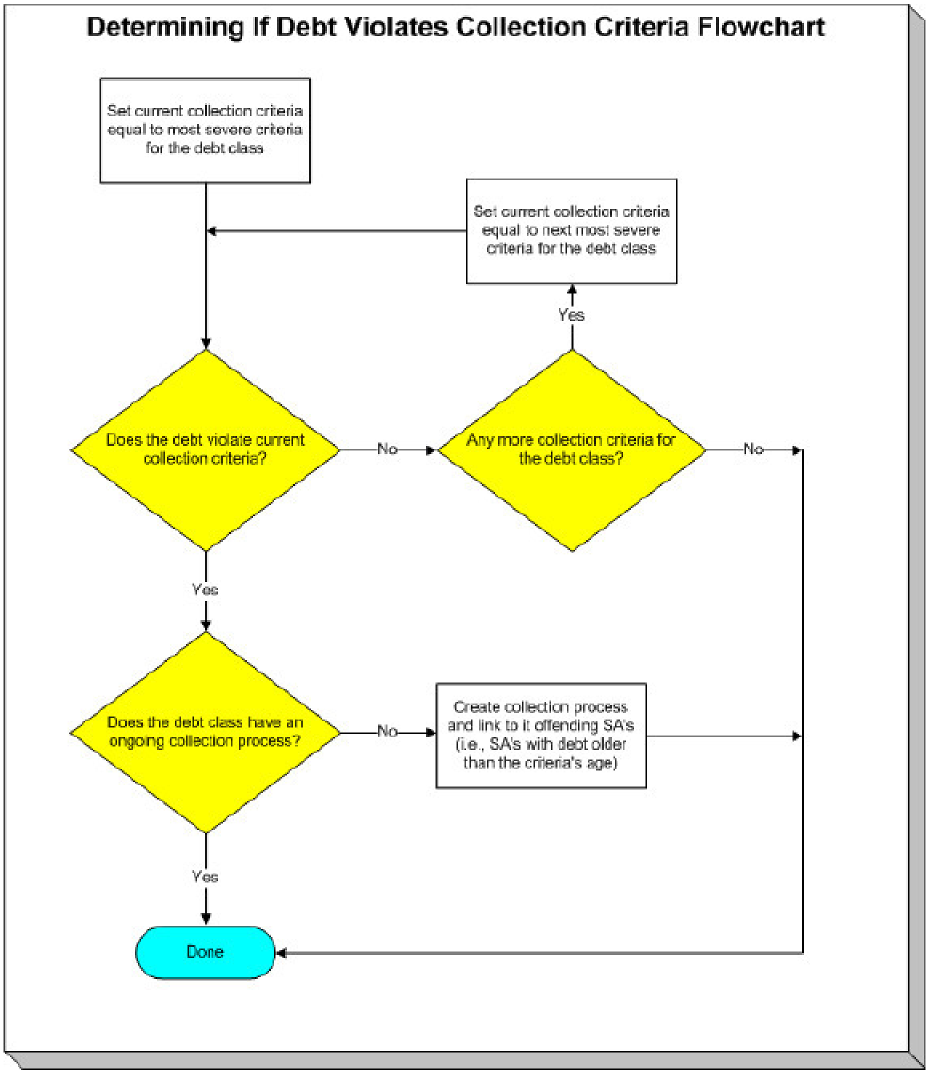
First, because an account belongs to a unique collection class, we only have to worry about a single column in the matrix when monitoring an account's debt.

Next, we accumulate the total amount of aged debt for each unique debt class associated with the account's service agreements.

Next, we subject the accumulated aged debt to the override aged debt algorithm (plugged in on the debt class). This algorithm can cause aged debt to be reduced. This is an optional algorithm and is only used if you set up pay plans for customers. Refer to How Pay Plans Affect The Account Debt Monitor for more information.

Next, we determine if the debt for the debt class violates the collection criteria in the respective matrix element. If so, we kick off a collection process and link the offending service agreements to it. The logic associated with the determination of whether to kick off a collection process is rather sophisticated. The following flowchart explains the exact details.

NOTE:
Important. If a service agreement is part of an ongoing severance process, it will NOT be considered by the Account Debt Monitor (it's already being severed). If a service agreement is stopped, closed, or reactivated, it also will NOT be considered by the Account Debt Monitor (it's already severed).


Determining If Debt Violates Collection Criteria Flowchart
Set current collection criteria equal to most severe criteria for the debt class
Set current collection criteria equal to next most severe criteria for the debt class
Yes
Does the debt violate current collection criteria?
No
Any more collection criteria for the debt class?
No
Yes
Does the debt class have an ongoing collection process?
No
Create collection process and link to it offending SA's (i.e., SA's with debt older than the criteria's age)
Yes
Done

NOTE:

> **Multiple collection processes may be kicked off.** It's important to be aware that if an account's service agreements reference multiple debt classes, a collection process will be started for each offending debt class.
>
> **One collection process per debt class.** A given debt class for an account may only have one ongoing collection process at any point in time.

What Happens When A Collection Process Is Started?

When you define collection criteria, you must define the collection process template to use if the criteria are violated. The system uses this template to create the account-specific collection process.

Every service agreement that is part of the offending debt class that has debt older than X days will be linked to the collection process (where X is the debt age on the collection criteria).

Also linked to the collection process will be one or more collection events. These events are typically a series of letters meant to prod the customer (you can also create an event that sends a To Do entry to a user to highlight the offensive debt). You define exactly which letters are generated and when they are generated when you set up the events on your collection process templates.

It's important to note that all of the collection events will be created when the collection process is created. Each of these collection events contains a trigger date. The trigger date of the first event(s) will typically be the current date. The trigger date of the other events will be in the future. Refer to Calendar vs Work Days for information that describes how the trigger date is set.

A separate process, Activate Collection Events, is responsible for activating collection events whose date is on or before the current date. Activation of an event causes the system to do whatever the event indicates (e.g., send a letter, send a To Do entry to a user, start a severance processes, etc.)

If adequate payments / credits are recorded in the system, the collection process will be cancelled.

> **FASTPATH:**
> For more information about collection process templates, see Setting Up Collection Process Templates. For more information about collection events, see The Big Picture Of Collection Events. For more information about how a collection process is cancelled, see How Are Collection Processes Cancelled.

Experimenting With Alternative Collection Process Templates

The system allows you to determine the efficacy of proposed collection process templates using a small subset of customers before implementing the templates on the entire customer base. We use the term "champion / challenger" to reference this functionality.

We'll use an example to explain. Let's assume your prevailing collection process template for residential customers starts with a "gentle reminder" letter followed 10 days later by a letter threatening collection agency referral if payment is not received. You may want to experiment with the impact of a change to this template. For example, you may want to change the "gentle reminder" to something more assertive and follow this up 5 days later with an even sterner warning. You can use the "champion / challenger" functionality to perform this experiment.

The following points describe how to implement "champion / challenger" functionality:

- Set up a "challenger" collection process template for each template that you want to experiment with.
- Insert a new **Champion/Challenger** option on the Collection Processing Feature Configuration for every champion template. Each option's value defines:

    - the "champion" collection process template code
    - the "challenger" collection process template code
    - the percentage of the time the system should use the "challenger" template
- Keep in mind that you can only experiment with one challenger template per champion template. For example, let's assume you have two prevailing collection process templates - one for residential customers and another for commercial customers. You can experiment with different challenger templates for the residential and commercial templates. However, you cannot experiment with two different challenger templates for the residential champion template (i.e., a champion template can have 0 or 1 challenger template).

After setting up the above, the Account Debt Monitor will use the challenger template X% of the time rather than the champion template.

If you are using the Oracle Utilities Analytics product, you can configure analytic zones in innumerable ways to compare the efficacy of the champion versus the challenger. For example,

- You can set up a graph to show the average duration of each type of process.
- You can set up a graph to show the average dollars that were successfully collected.
- You can set up a dimensional scorecard to show how each template performed in different regions (or customer classes or ...).
- Etc (the list is limited by your imagination.

How Are Collection Processes Cancelled?

A collection process may be cancelled via the mechanisms described in this section.

The Collection Process Monitor Can Cancel A Collection Process

The Collection Process Monitor (CPM or C1-CPM)) is a background process that reviews a collection process when the debt associated with one of its service agreements is reduced. Financial events that can cause service agreement debt to be reduced are:

- The cancellation of a bill segment.
- The creation of a payment segment.
- The creation of an adjustment that credits a service agreement.

The review performed by the CPM occurs as follows:

- **Debt class cancel criteria.** In general, the sum of all debt associated with the collection process's debt class must be less than a given threshold amount for a collection process to be cancelled. If so, the collection process is cancelled.
- Please be aware that, if a Pay Plan exists for the account and debt class, the customer's debt will be temporarily reduced by the amount of the pay plan's scheduled payments before it is compared to the threshold amount. Please be aware that this temporary reduction will only occur if you have plugged in the appropriate pay plan debt reduction algorithm on the debt class.

> **NOTE:**
> **The above logic is not "hard coded".** The CPM calls the Collection Process Cancel Criteria Algorithm defined on the debt class that is associated with the collection process. This algorithm will cancel a collection

process if the sum of ALL service agreements in the debt class have debt less than a given threshold amount. However, because it's an algorithm, you can introduce whatever cancellation criteria you please.

- **Service agreement cancel criteria.** You can optionally introduce a special quirk to the cancellation logic. This quirk is a bit difficult to understand. To understand it, you should recall:
    - All service agreements that are in arrears in a given debt class are linked to the collection process.
    - The collection event called Start Severance creates a severance process for every service agreement that is in arrears on the collection process (the alternative is to Nominate A Single Service Agreement To Sever).
    - If you use the Start Severance collection event, you would want to remove a service agreement from a collection process when it no longer has intolerable debt (regardless of the state of the debt class's entire debt). You'd want to do this because, if you don't, the system would start a severance process for the paid up service agreement and if it's paid up, you wouldn't want a severance process created for it.
- To "remove" service agreements from a collection process when they no longer have intolerable debt, you should plug-in a Service Agreement-Oriented Cancel Criteria Algorithm on your collection process templates. The CPM will call this algorithm if you've plugged it in.

➤ **NOTE:**
When all service agreements are "removed" from a collection process, the CPM cancels all pending collection events and cancels the collection process.

WARNING:
Checking if individual service agreements should be removed from a collection process is optional (meaning that you don't have to plug one in on the collection process template).

A New Payment Plan Can Cancel A Collection Process

Refer to Collection Process / Severance Process Cancellation When A Pay Plan Is Created for the details.

➤ **NOTE:**
Real time cancellation. Please be aware that the system will cancel a collection process real time when a pay plan is created (if the pay plan's scheduled payments are enough to pay-off the customer's outstanding debt).

A User May Cancel A Collection Process At Their Discretion

A user may cancel a collection process at their discretion.

Stopping A Service Agreement May Cancel A Collection Process

The system will "remove" a service agreement from a collection process when it is stopped (i.e., when the service agreement's status becomes Stopped). When the last service agreement is "removed" from the collection process, the collection process will be cancelled.

The Big Picture Of Collection Events

This section describes the various types of collection events and their lifecycle.

How Are Collection Events Created?

Collection events may be created as follows:

- The Account Debt Monitor creates a collection process when an account violates collection criteria. The collection process has one or more collection event(s). The number and type of events is controlled by the collection process template associated with the collection process.
- Collection events are created when a user creates an ad hoc collection process. The number and type of events is controlled by the collection process template defined when the collection process is created.
- An ad hoc collection event may be created and linked to an existing collection process by a user at their discretion.

➤ **NOTE:**
Bottom line. Most collection events are created by the system when it creates a collection process for delinquent accounts. If you need to create an ad hoc collection event, you can either create a collection process whose template contains the desired event OR link the desired event to an existing collection process.

● **FASTPATH:**
For more information about the creation of events by the Account Debt Monitor refer to What Happens When A Collection Process Is Started?. For more information about creating ad hoc collection processes, refer to Collection Process Maintenance. For more information about creating ad hoc events, refer to Collection Process - Events.

Types Of Collection Events

The following table describes the various types of collection events and what happens when they are completed:

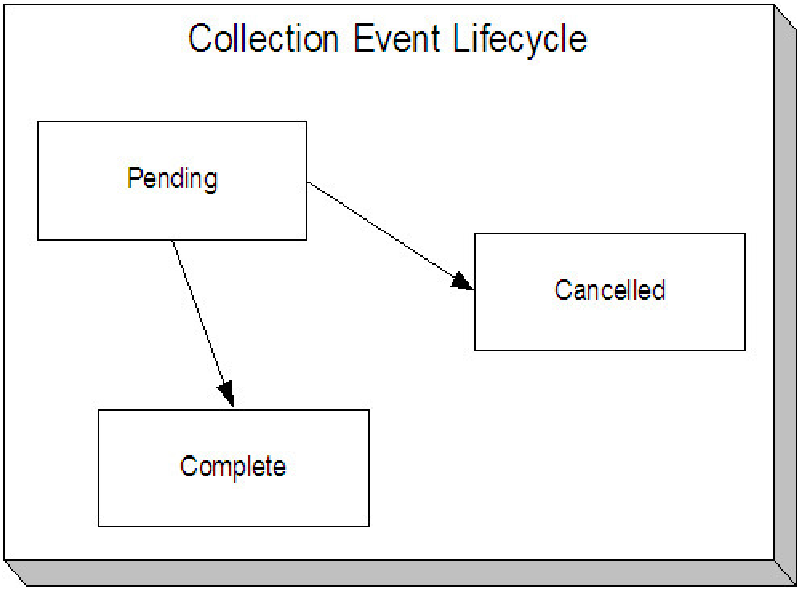
Type Of Collection Event	What Happens In The System
Send Letter	A customer contact is created for every financially responsible person linked to the account. The customer contact causes a letter to be produced. The type of letter is defined on the customer contact's contact type. The recipient of each letter is defined on Account / Person (those persons marked as **Receiving Notifications**).

Create To Do Entry	A To Do entry is created. Refer to The Big Picture of To Do Entries for more information about To Do entries.
Affect Credit Rating/Cash-Only	An account credit rating demerit record is created. The number of demerits is defined on the collection event type.
Cancel Budget	Every service agreement linked to the account that is on a budget is "removed" from the budget (i.e., the recurring charge amount for the service agreement is set to zero). In addition, a syncing adjustment is issued to cause each SA's current balance to be set equal to their payoff balance (the adjustment type is defined on the SA's SA type). Also, the Budget Plan field on the Account - Budget page is cleared.
Generic Algorithm	The algorithm defined on the event's event type is executed.
Start Severance Process	A severance process is created for every service agreement linked to the collection process. The type of severance process is defined on the SA type table (every service agreement references an SA type).

- **FASTPATH:**
  Refer to Setting Up Collection Event Types for more information.

Collection Event Lifecycle

The following diagram shows the possible lifecycle of a collection event:



Collection events are initially created in the pending state.

When the system sees a pending event with a trigger date on or before the current date, the system executes the event's activity and completes the event.

- **FASTPATH:**

For more information about a collection event's trigger date, see Collection Event Trigger Date.

A pending event will be cancelled automatically by the system when the account's debt no longer violates the collection criteria that sparked the event's collection process. A pending event may also be cancelled by a user at their discretion. Refer to How Are Collection Processes Cancelled for more information about how the system will cancel a collection process (and its events).

Collection Event Trigger Date

When a collection event is created by the system, its trigger date is set in accordance with your date arithmetic preferences. Refer to Calendar vs. Work Days for more information.

How Are Collection Events Completed

A background process runs periodically (at least daily) that looks for collection events with a trigger date on or before the current date. For each triggered event, the system executes its activity and then completes it. Refer to Collection Event Activator for more information.

The Last Collection Event Should Kick Off Severance Process(es)

The last collection event will typically kick off the severance process for every service agreement linked to the collection process. This will only happen if you set up the collection process template accordingly (i.e., the last event type in the process template is the kind that starts a severance process for every service agreement linked to the collection process).

➤ **NOTE:**

Nominating a service agreement to sever. Many organizations that have multiple services in arrears will NOT sever every service agreement that's in arrears. Rather, they will nominate one service agreement and use it to encourage the customer to pay for the other services. If your organization works this way, then your last collection event should call the Nominate A Service Agreement To Sever Algorithm.

How Are Collection Events Canceled?

Users can cancel a collection event at their discretion. In addition, the system can cancel a collection event when it automatically cancels a collection process. Refer to How Are Collection Processes Cancelled for the details.

The Big Picture Of Severance Process Cancellation

The topics in this section provide high level information about the cancellation of severance processes.

● **FASTPATH:**

For more information refer to The Lifecycle Of A Severance Process And Its Events.

How Are Severance Processes Cancelled?

A severance process may be cancelled via the mechanisms described in this section.

FASTPATH:
Refer to What Happens When A Severance Process Is Cancelled? for what happens when a severance process is cancelled.

The Freezing Of Certain Financial Transactions Can Cancel A Severance Process

NOTE:
The system will only cancel a severance process if its severance process template indicates that **Auto Cancel** is allowed. Typically, this switch is set on all severance process templates except for the odd ones that are used to reconnect service.

The system reviews a severance process real-time whenever its service agreement's debt is reduced. Financial events that can cause service agreement debt to be reduced are:

- The cancellation of a bill segment.
- The creation of a payment segment.
- The creation of an adjustment that credits a service agreement.

NOTE:
Real time cancellation. Unlike collection processes, the system cancels severance processes real time (i.e., there is no background process that monitors severance processes). Why are severance processes canceled real time? Because a severance process may have events that create field activities to sever service. These events need to be canceled the moment the FT is frozen, we can't wait until a background process runs. This means that if a customer pays in person for a service agreement that is pending severance, the system will cancel the process and its field activities (if any) the moment the payment is entered.

The review takes place as follows:

- **Debt class cancel criteria.** In general, the sum of all debt associated with the severance process's debt class must be less than or equal to a given threshold amount for a severance process to be cancelled. If so, the severance process is cancelled.
- Please be aware that, if a Pay Plan exists for the account and debt class, the actual debt will be temporarily reduced by the amount of the pay plan's scheduled payments before it is compared to the threshold amount. Note: this temporary reduction will only occur if you have plugged in the appropriate pay plan debt reduction algorithm on the debt class.

➤ **NOTE:**

The above logic is not "hard coded". The system calls the Severance Process Cancel Criteria Algorithm defined on the debt class that is associated with the severance process. This algorithm cancels a severance process if the sum of ALL service agreements in the debt class have debt less than or equal to a given threshold amount. However, because it's an algorithm, you can introduce whatever cancellation criteria you please.

- **Service agreement cancel criteria.** You can optionally introduce a special quirk to the cancellation logic. This quirk is a bit difficult to understand. To understand it, you should recall:
  - The collection event called Start Severance creates a severance process for every service agreement that is in arrears. Note: you would only use this type of collection event if you do not Nominate A Single Service Agreement To Sever.
  - If you use the Start Severance collection event, then you would want to cancel a severance process when its service agreement no longer has intolerable debt (regardless of the state of the debt class's entire debt).
- To cancel a severance process when its related service agreement no longer has intolerable debt, you should plug-in a Cancel Criteria Algorithm on your severance process templates. The system will call this algorithm if you've plugged it in.

➤ **NOTE:**

Manual Creation. A user can create a severance process for an account that does not qualify to be on severance according to the cancel criteria algorithm. For example, perhaps your cancel criteria algorithm cancels a severance process when the account's debt falls below a threshold amount. A user can create a severance process for an account whose debt is already below this threshold. Because cancellation is real time, there is no action that will cause this severance process to be canceled. When a manual severance process is created, the system executes the appropriate cancellation criteria algorithm. If the algorithm indicates that the system would have canceled this severance process, a warning is issued.

A New Payment Plan Can Cancel A Severance Process

Refer to Collection Process / Severance Process Cancellation When A Pay Plan Is Created for the details.

➤ **NOTE:**

Real time cancellation. Please be aware that the system will cancel a severance process real time when a pay plan is created that pays off enough debt.

A User May Cancel A Severance Process At Their Discretion

A user may cancel a severance process at their discretion.

Stopping A Service Agreement Will Cancel A Severance Process

The system will cancel a severance process if its service agreement is stopped (i.e., when the service agreement's status becomes Stopped).

What Happens When A Severance Process Is Cancelled?

The following takes place when a severance process is canceled by the system:

- The system cancels all pending severance events and deactivates the severance process.
- If there are any field activities linked to the severance process, an optional plug-in spot defined on the installation record allows you to plug in an algorithm to cancel these field activities.
    - The base package Severance Process Cancellation Algorithm will cancel all pending field activities that were created as a result of the severance process that are not linked to a dispatched field order.
- If there are any pending field activities left associated with the severance process, it is marked to trigger the creation of a To Do entry to highlight that field activities exist for a canceled severance process. (This happens if you have not plugged in an algorithm to perform the cancellation or if the algorithm detected a condition that prevented cancellation.) To create the To Do entry, you must run the background process TD-SPRO.
- There is an optional plug-in spot defined on the severance process' template. If an algorithm is plugged-in, it is called. The base package algorithm will create a reconnect process if there are completed field activities for a cut for nonpayment severance event associated with the severance process. Refer to Severance Post Cancellation Algorithm for more information about this algorithm.

The Big Picture Of Severance Events

This section describes the various types of severance events and their lifecycle:

How Are Severance Events Created?

Severance events may be created as follows:

- The process that completes (i.e., executes) collection events creates a severance process when it completes a "start severance process" collection event. The severance process has one or more severance event(s). The number and type of events is controlled by the severance process template associated with the severance process. Refer to The Collection Event Activator for more information about this process.
- Severance events will be created when a user creates an ad hoc severance process. The number and type of severance events is controlled by the severance process template associated with the severance process.
- An ad hoc severance event may be created and linked to an existing severance process by a user at their discretion.

NOTE:
Bottom line. Most severance events are created by the system when it creates a severance process for delinquent service agreements. If you need to create an ad hoc severance event, you can either create a severance process whose template contains the desired event OR link the desired event to an existing severance process.

FASTPATH:

For more information about creating ad hoc severance processes and events, refer to How To Perform Common Severance Process Functions.

Types Of Severance Events

The following table describes the various types of severance events and what happens when they are completed:

Type Of Severance Event	What Happens In The System
Send Letter	A customer contact is created for every financially responsible person linked to the service agreement's account. It is the customer contact that causes a letter to be produced. The type of letter is defined on the customer contact's contact type. The recipient of each letter is defined on Account / Person (those persons marked as **Receiving Notifications**).
Create To Do Entry	A To Do entry is created. Refer to The Big Picture of To Do Entries for more information about To Do entries.
Create Field Activities	A field activity is created for each service point associated with the service agreement being severed. The type of activity is defined on the service point's SP type's field activity type profile.
Generic Algorithm	The algorithm defined on the event's event type is executed.
Expire Service Agreement	The service agreement is expired and, if earlier severance events created "cut for non-payment" field activities, these field activities will be used as the basis for stopping service. Refer to Finalizing Pending Stops for how the system use the meter reads on these field activities as the "stop reads" on the service agreement. Note, you can see the field activities that are used to "cut" and "stop" service by viewing the Field Activities grid on Service Agreement - Service Point.
Affect Credit Rating/Cash-Only	An account credit rating demerit record is created. The number of demerits is defined on the collection event type.

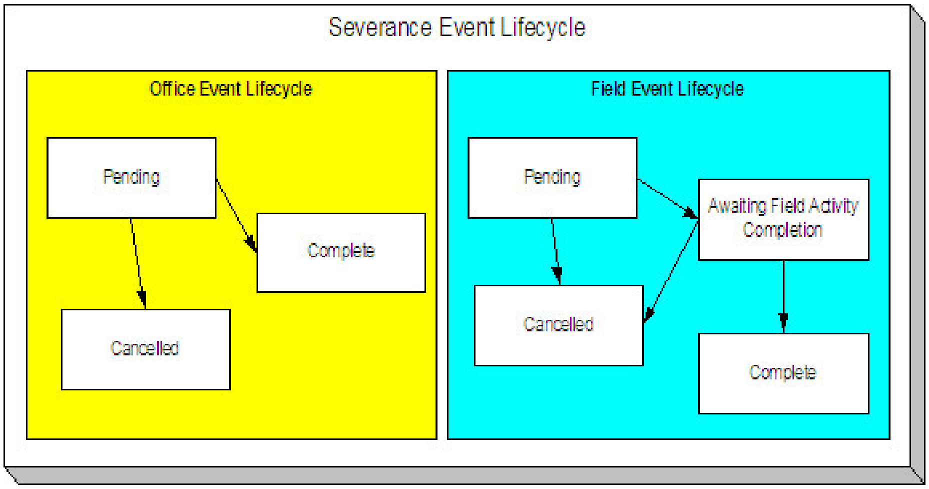
FASTPATH:
Refer to Setting Up Severance Event Types for more information.

Field Events Versus Office Events

Severance events are considered either field or office events. Office events are just like collection events in that they don't involve any type of field activity in their completion. Field events, however, are not like collection events because they create one or more field activities. These activities cause these events to have a different lifecycle. Refer to Severance Event Lifecycle for more information.

Severance Event Lifecycle

The following diagram shows the possible lifecycle of a severance event:



The following points explain the lifecycle of severance events of the office variety:

- Office events are initially created in the pending state.
- A pending office event becomes complete when the system sees that its trigger date is on or before the current date. At this time, the system executes the event's activity (e.g., create a letter, create To Do entry).

  - **FASTPATH:**
    For more information about a severance event's trigger date, see Severance Event Dependencies & Trigger Date.

- A pending office event will be cancelled automatically by the system when the debt associated with the severance process's service agreement is sufficiently reduced. A pending office event may also be cancelled by a user at their discretion. Refer to How Are Severance Processes Cancelled for more information about how the system will cancel a severance process (and its events).

The following points explain the lifecycle of severance events of the field variety:

- Field events are initially created in the pending state.
- A pending field event becomes awaiting field activity completion when the system sees that its trigger date is on or before the current date. At this time, the system creates the field activities associated with the given event (e.g., disconnection warning, disconnect for nonpayment, etc.).

  - **FASTPATH:**
    For more information about a severance event's trigger date, refer to Severance Event Dependencies & Trigger Date. For more information about the field activities that are created for a field severance event, refer to Field Events And Their Activities.

- An awaiting field activity completion field event becomes complete when the system sees that its field activities are all complete or cancelled.
- A pending field event will be cancelled automatically by the system when the service agreement associated with the severance event's severance process has sufficient credits. A pending field event may also be cancelled by a user at their discretion.
- An awaiting field activity completion field event will be cancelled automatically by the system when the service agreement associated with the severance event's severance process has sufficient credits if the field activity has not been dispatched (refer to Designing Your Reconnection Procedures for information on how the system handles the situation if the field activity is completed or dispatched). An awaiting field activity completion event may also be cancelled by a user at their discretion.

Severance Event Dependencies and Trigger Date

When a severance event is created by the system, its trigger date cannot be set. This is because, unlike collection events, the trigger date on severance events can only be set when ALL of the preceding severance events on which it depends are complete. An example will help explain why this design is necessary. Consider the following example that shows a standard severance process and its events:

Event Number	Severance Event	Dependent On Event(s)	Trigger Date Set To X Days After Completion Of Preceding Events
10	Field activity - Disconnect for non-payment warning	N/A - first event	0
20	Field activity - Disconnect for non-payment	10	2
30	Create To Do entry	20	0
40	Send letter to customer	20	0
50	Expire service agreement	20	10

This severance process is meant to execute as follows:

- On the first day, generate a 48-hour warning of impending disconnection. This is a field event as most organizations deliver this warning in person.
- After 2 days (i.e., 48 hours) have passed, create a field activity to disconnect for non-payment.
- When this is completed, generate a To Do entry to let a CSR know about the cutoff. Also, send a letter to the customer.
- If after 10 days from the cutoff, we still don't have payment, expire the service agreement.

As can be seen from the above example, the later events are dependent on the completion of the field activities in the earlier events. This means that when you set up a severance event, you must indicate the events on which it depends (and the number of days after their completion that the event should be triggered).

➤ **NOTE:**

Bottom line. The system sets the trigger date on severance events when it detects that all of its dependent events are complete (this is the responsibility of the SED or C1-SED background process). Refer to Calendar vs Work Days for a description of your choices in respect of how the trigger date is calculated.

Field Events And Their Activities

When the system is told to start a severance event that creates field activities (e.g., disconnect for non-payment), it will generate a field activity for every service point linked to the service agreement.

The question is, where does it get the field activity type associated with the field activities? The answer is explained below:

- Every service point has an SP type.
- Every SP type references a field activity type profile.
- A field activity type profile contains a matrix defining activity types to generate under various situations. Some of these situations are those associated with severance.

> ●
>
> **FASTPATH:**
> For more information, refer to Setting Up Field Activity Type Profiles.

Severance Event Activation & Completion

A background process runs periodically (at least daily) that looks for severance events with a trigger date on or before the current date. This process executes the activity associated with each event. If the event is an office event, the event then becomes complete. If the event is a field event, the event becomes "awaiting field activity completion" until the field activities are complete. At this time, the severance event is completed. Refer to Severance Event Activator for more information.

How Are Severance Events Canceled?

Users can cancel a severance event at their discretion. In addition, the system can cancel a severance event when it automatically cancels a severance process. Refer to How Are Severance Processes Cancelled for the details.

The Big Picture Of Write Off Processing

Before you financially write-off debt, most companies go to some effort to collect the past due funds. You control exactly what happens by setting up the various write-off control tables. The topics in this section provide background information that will help you understand how the information in these control tables is used.

How Is Debt Financially Written-Off?

Before debt can be written-off, a write-off service agreement must exist for the account. Why? Because when you write-off a normal service agreement's debt, you are actually transferring its debt to a write-off service agreement.

A write-off service agreement is just like other service agreements in that:

- It holds debt.

- When a payment is received, the service agreement's debt is reduced.

Debt is transferred to a write-off service agreement (WO SA) from the customer's uncollectable service agreements (SAs). The following points highlight important characteristics about the uncollectable SAs and the WO SA:

- The WO SA and the uncollectable SAs should be linked to the same account (note: this isn't a strict rule, it just makes sense because an account's written off funds should be linked to the account).
- Debt may be transferred to a WO SA from any type of service agreement regardless of debt class, i.e., a WO SA can contain debt that originated in any debt class.
- When you transfer debt from the uncollectable SAs to the WO SA, the debt is removed from the uncollectable SAs (and their status becomes closed - assuming their balance becomes zero).
- If you use the system's automated write-off processing, the system will create WO SAs for you. The system's automated write-off processing can write-off revenue in a different manner than is used to write-off liabilities. Refer to The Ramifications of Write Offs in the General Ledger for more information.
- WO SAs are immune from the account debt monitor (assuming their debt class is marked as not being subject to collection activities).
- WO SAs are not billed (assuming their SA type is marked as being not billable).
- WO SAs start their life with a non-zero payoff and current balances (i.e., they have debt when first started). This debt is transferred from the normal service agreement(s) whose uncollectable debt necessitated the creation of the WO SA.
- If the customer pays off the write-off debt, the WO SA remains active in case you ever need to write-off debt in the future. If you don't like the WO SA remaining active after it's paid off, you can indicate on the WO SA's SA Type that it is a "one time charge", this will cause the WO SA to be automatically closed when it's paid off.
- You can transfer additional uncollectable debt to the WO SA.

➤ **NOTE:**
Bankruptcy write-offs. If you have to write-off debt because a customer declares bankruptcy, everything stated above is true. The only thing you have to do is use a different SA type for bankruptcy write-offs as compared to "normal" write-offs. On the bankruptcy write-off SA type, simply leave the payment segment type blank - this way the system will never distribute a payment to the bankrupt debt (because bankrupt debt is legally uncollectable).

The Ramifications of Write Offs in the General Ledger

WARNING:
If you practice cash accounting, refer to Cash Accounting and Write-Offs.

When you write-off unpaid debt, you shouldn't book it all to a write-off expense account. Why? Because the debt that you're writing off typically contains both revenue and liabilities. At write-off time, you typically want to:

- Book the written off revenue to a write-off expense account, and
- Reduce the liabilities (you don't owe the liability if you don't get paid).

Consider the following example of a simple electric service agreement with two financial transactions:

Event	GL Accounting
Customer is billed	A/R 1000
	Revenue <900>

	State Tax Payable - Taxing State - California <80>
	City Tax Payable - Taxing City - San Francisco <20>
Customer is levied a late payment charge	A/R 50
	Late Payment Revenue <50>

After these two financial transactions are booked, the customer has debt of $1050. Of this $1,050; $950 is revenue and $100 is liability (money you owe the taxing authorities).

If the customer doesn't pay, you will eventually have to write-off this debt. Most organizations would issue the following types of financial transactions to do this:

Event	GL Accounting
Write-off the bill	Write-off Expense 900
	State Tax Payable - Taxing State - California 80
	City Tax Payable - Taxing City - San Francisco 20
	A/R <1000>
Write-off the late payment charge	Write-off Expense 50
	A/R <50>

Notice in the above transactions, the two separate revenue accounts are written off by booking to an expense account. However, the liability accounts are reversed. Why is revenue treated differently from liabilities at write-off time? There's a good reason for it (if you're an accountant), for the time being, just accept that this is how it works.

And finally, we need to worry about what happens if the customer eventually pays off his written off debt. If this happens, most organizations would pay off the write-off first, and, if there was still money left, they'd reimburse the taxing authorities. If we assume the customer pays off the entire written off debt, the following financial transactions would be issued:

Event	GL Accounting
Pay off the written off debt	Cash 900
	Write-off Expense <900>
Reinstate the liabilities	Cash 100
	State Tax Payable - Taxing State - California <80>
	City Tax Payable - Taxing City - San Francisco <20>

While the reinstatement of liabilities at payment time is possible in the system, the ramifications of doing such make this approach impracticable (the ramifications are a) if the check bounces, we would not be able to reduce the liabilities, and b) if there was a partial payment of the liabilities, the remaining unpaid amount could get written down). Therefore, when a write-off is paid the following financial transactions should be issued:

Event	GL Accounting
Pay off the written off debt	Cash 900
	Write-off Expense <900>
Reinstate the liabilities	Cash 100
	Reinstated liabilities <100>

Notice that rather than reinstating the individual liabilities, we simply reinstate all liabilities into a single account. This means your accountants will have to distribute this money to the appropriate liabilities manually.

So, how do we achieve the above in the system? This explanation is a little complicated, but it'll make sense if you keep the above financial transactions in mind:

- First of all, you'll need two different SA types - one to hold the written off revenue and another to hold the reduced liabilities.
    - On the SA type that holds written off revenue, indicate that it is not billable, indicate that it cannot have excess credits, and give it a high payment distribution priority. The distribution code on this SA type should reference your Write-off Expense account.
    - On the SA type that holds the reduced liabilities, indicate that it is not billable, indicate that it cannot have excess credits, and give it a high payment distribution priority. The distribution code on this SA type should reference a the "reinstated liabilities" GL account.

Next, you need to understand how the system's standard write-off logic works:

- The system accumulates the distribution codes from GL details associated with recent financial transactions linked to the service agreement being written-off.
- When the system has accumulated enough distribution codes (i.e., where the amount associated with the distribution code equals or exceeds the amount to write off), the debt will be transferred to a new or existing write-off service agreement(s). The number and type of service agreements to which the bad debt is transferred is defined on the distribution codes. Refer to Setting Up Distribution Codes for how to define the type of write-off service agreement associated with a distribution code. In our example, we'd need the two SA types described above - one for the revenue accounts, the other for the liability accounts.
- At write-off time, for those distribution codes associated with revenue, the system will create a transfer adjustment from the normal service agreement to the write-off revenue service agreement. This will reduce (credit) the receivable on the normal service agreement and increase (debit) the expense account defined on the write-off revenue service agreement.
- However, if we do the above for the distribution codes associated with liabilities, we have a problem. The problem is a bit hard to explain unless you understand tax accounting, but it basically comes down to this - if we simply transfer the portion of the receivable balance associated with the liabilities to the write-off liability SA, we will always be debiting the distribution code defined on the SA type. This isn't correct because we really want to debit the liability account (and reference the characteristic type and value from the original credit) when we reduce the liability. So how do we do this? For those distribution codes associated with liabilities, you need to indicate that you want to override the distribution code on the "transfer to" side of the transfer adjustment with the distribution code / characteristic type / characteristic value that was originally booked. Refer to Setting Up Distribution Codes for how to indicate you want to override the distribution code at write-off time. If you do the above, then at write-off time the transfer adjustment will reduce (credit) the receivable on the normal service agreement and increase (debit) the original liability accounts from the original financial transactions.

If you followed the above, you'll see that we now have everything debited and credited appropriately. And, if a payment materializes for the written off debt, we will simply debit cash and credit the distribution code on the respective SA (either Write Off Expense or Reinstated Liabilities).

NOTE:
Batch and real-time write-offs may use the above processing. The above logic is executed real time when a user writes off debt using the write-off transaction (assuming the base package write off algorithm is plugged into the account's customer class). The above logic is executed in batch when a write-off event that references a Write Off Using Distribution Codes event type is executed. Write-off events are described in detail below.

Automated versus Manual Write Offs

The system will automatically create write-off SAs and transfer uncollectable debt to them during the automated write-off processing described below.

If necessary, you can write-off debt outside of the automated write-off process using either of the following methods:

- You can transfer bad debt from any service agreement to a write-off service agreement using a transfer adjustment.
- You can use the write-off transaction to write-off debt real-time. When this transaction is used, the system executes the logic embedded in the Write Off Method algorithm that's plugged in on the account's customer class.

How Does The Write-Off Monitor Work?

This section describes how the Write Off Monitor uses your write-off criteria and write-off process templates to collect overdue debt.

Different Write-Off Criteria For Different Customers And Different Debt

Consider the following:

- You probably have different write-off criteria for different customer segments. For example, customers with large bills probably have strict criteria, whereas you're probably more lenient with small customers (or vice versa). You differentiate your customers in respect of the collection process via a **collection class code on the customers' accounts**. An account's initial collection class is defaulted from its customer class. You may override an account's collection class at will.
- You probably have different write-off criteria for different classes of debt. For example, if a customer has both regulated and unregulated debt, you probably have commission-imposed criteria to be applied to the regulated debt, but you have control over how to write-off unregulated debt. You differentiate your debt in respect of the collection process via a write-off **debt class on the customers' service agreements** (note the write-off debt class is actually defined on the SA type and every service agreement has a SA type).

➤ **NOTE:**
Write Off Debt Class vs. Regular Debt Class. It's important to be aware that a SA type references both a regular debt class and a write-off debt class. The regular debt class controls the collection criteria applied against an account's service agreements. The regular debt class is also used to segregate an account's outstanding balance on several queries in the system. The write-off debt class controls the write-off criteria applied against an account's stopped service agreements. The reason the system supports two different debt classes is because you may categorize your service agreements differently when you try to collect overdue debt versus when you write-off debt.

Given the above, you should understand that different write-off criteria will exist for every combination of collection class and write-off debt class. If you're confused, then consider the following matrix:

SA's Write-Off Debt Class	**Account's Collection Class:**	**Account's Collection Class:**
	Commercial Customer	**Residential Customer**
Regulated	N/A - there is no regulated, commercial customer debt.	Attempt to reduce the SA's balance to zero using the following methods: Synchronize current balance with payoff balance.

		Attempt to transfer debt to another active service agreement linked to the account. If the debt is < $10 and > $-1, write down the debt using a write-down adjustment. If the debt is <= $-1, create an A/P adjustment to refund the credit to the customer. If debt remains, create the default write-off process for regulated debt.
Unregulated	Attempt to reduce the SA's balance to zero using the following methods: Synchronize current balance with payoff balance. Attempt to transfer debt to another active service agreement linked to the account. If the debt / credit is < $10 and > $-10, write down the debt using a write-down adjustment. If the debt / credit is <= $-10, create an A/P adjustment to refund the credit to the customer. If debt still remains: Highest priority: If customer has a non-cash deposit, create the non-cash deposit write-off process. Otherwise, create the default write-off process for unregulated commercial debt.	Attempt to reduce the SA's balance to zero using the following methods: Synchronize current balance with payoff balance. Attempt to transfer debt to another active service agreement linked to the account. If the debt is < $10 and > $-1, write down the debt using a write-down adjustment. If the debt is <= $-1, create an A/P adjustment to refund the credit to the customer. If debt remains, create the default write-off process for unregulated residential debt.

Notice that each cell in the matrix has the same pattern:

- The system first attempts to reduce the SA's current and payoff balances to zero using the following methods (assuming you have set up the write-off control appropriately):
  - Sync the current balance with the payoff balance. If the SA's payoff balance is zero, this will cause the current balance to become zero and therefore close the SA.
  - If there's a debit balance, transfer the debt to any pending start or active SA in the same write-off debt class.
  - If there's a credit balance, transfer the debt to any non-closed / non-cancelled SA in the same write-off debt class
  - If the remaining debit / credit balance is within a user-defined tolerance (this is defined on the respective algorithm on the write-off control), create an adjustment to write-down the small balance.
  - If a credit balance remains, create an A/P adjustment to refund the balance with a check (the adjustment type is defined on the respective algorithm on the write-off control).
- All of the above points will cause the SA to close. If debt remains, the system starts some type of write-off process. The type of process is dependent on the respective criteria. What differentiates one write-off criteria from another is its priority. The higher priority criteria will be compared first. If the customer / debt meets the criteria, the write-off process is initiated; no further comparisons are performed.

FASTPATH:
For more information about maintaining this matrix, refer to Setting Up Write-off Control.

When Is Debt Monitored For Write Off Purposes?

The write-off monitor only reviews a service agreement when the following conditions are true:

- The service agreement is stopped and reactivated.
- If the service agreement is a "billable charge" SA (as identified on its SA type), all of its billable charges must appear on a bill segment AND the bill segment's bill's due date plus grace period must be on or before the business date.
- If the service agreement is not a "billable charge" SA AND it is billable (as identified on its SA type), the SA must have a closing bill segment (i.e., it must be final billed) and the bill segment's bill's due date plus grace period must be on or before the business date.
- If the service agreement is a sub SA, its master SA must abide by the above conditions.
- If the service agreement is not billable, it is possible that adjustments, which affect the SA's debt, exist. The write-off monitor will only review a non-billable SA if all FTs for this SA that have been marked to include on a bill have been swept onto a bill and the bill for any of these FTs has a bill due date plus grace period on or before the business date.

> **NOTE:**
> **Postponing write-off processing.** You can prevent the write-off process from processing an eligible service agreement by populating the account's C&C Postpone Date with a future date.

Attempt To Close The SA Before Creating A Write Off Process

Before the write-off monitor creates a write-off process for a stopped and reactivated service agreement, it attempts to reduce the service agreement's debt to zero using all of the following methods:

- If the account has active service agreements, it will transfer the finaled debt to a pending start or active service agreement.
- If the debt or credit amount on the service agreement is small, the system will generate an adjustment to 'write it down' (or up in the case of a small credit).
- If the service agreement has a large credit amount, the system will generate an A/P adjustment (resulting in a check being sent to the customer).

> **NOTE:**
> **Plug-in algorithms do the work.** Algorithms that are plugged-in on the write-off control responsible for managing the service agreement's debt actually perform the above effort. You can customize these algorithms to behave exactly how your collections staff desires.

If the algorithms responsible for the above effort are successful in reducing the service agreement's debt to zero, then the service agreement closes and will not be subject to write-off processing. If the above algorithms don't result in the service agreement's debt being reduced to zero, a write-off process will be started (as describe below).

What Happens When A Write-Off Process Is Started?

When you define write-off criteria, you must define the write-off process template to use if the criteria are violated. The system uses this template to create the account-specific write-off process.

Every stopped or reactivated service agreement that is part of the offending write-off debt class will be linked to the write-off process.

Also linked to the write-off process will be one or more write-off events. These events are meant to prod the customer. You define the types of events and when they are generated when you set up your write-off process templates.

It's important to note that all of the write-off events will be created when the write-off process is created. Each of these write-off events contains a trigger date. The trigger date of the first event(s) will typically be the current date. The trigger date of the other events will be in the future. Refer to Calendar vs Work Days for a description of how the trigger date is calculated.

A separate process, Activate Write-off events, is responsible for activating write-off events whose date is on or before the current date. Activation of an event causes the system to do whatever the event indicates (e.g., send a letter, send a To Do to an operator, refer debt to a collection agency, etc.)

➤ **NOTE:**
Multiple write-off processes may be kicked off. It's important to be aware that if an account's service agreements reference multiple write off debt classes, a write-off process will be started for each offending write off debt class that has stopped or reactivated service agreements.

● **FASTPATH:**
For more information about write-off process templates, see Setting Up Write Off Process Templates. For more information about write-off events, see The Big Picture Of Write-off Events. For more information about how a write-off process is cancelled, see How Does A Write-Off Process Get Cancelled?.

How Does A Write-Off Process Get Cancelled?

The system "removes" a service agreement from a write-off process when its status becomes closed (i.e., when its balance is zero). When all service agreements are removed, the system cancels all pending write-off events and deactivates the write-off process. When the write-off process is deactivated, all collection agency referrals associated with the write-off process are cancelled.

➤ **NOTE:**
Removing closed service agreements from a write-off process. Service agreements aren't actually removed from the process. Rather, they are inactivated so a proper audit exists.

How Do Collection Agency Referrals Work?

The following points describe how collection agency referrals work.

- A write-off process has one or more events. One type of event causes overdue debt to be referred to a collection agency.
- When a referral write-off event is activated, the system marks the event for processing by the event's Collection Agency Referral Algorithm (refer to Setting Up Write Off Event Types for more information).

- The next time the Collection Agency Referral process executes (the frequency is dependent on your background process schedule), it will refer the process' debt to a collection agency. The specific agency to which the debt is referred is controlled by the event type's Collection Agency Referral Algorithm. The sample algorithm supplied with the system simply refers debt to the collection agency with the least amount of referred debt. If you prefer different logic, you must write your own algorithm.
- Regardless of the manner in which a collection agency is selected for an account's debt, the referral involves the creation of a collection agency referral history record. Refer to Collection Referral for more information.
- A collection agency referral history record is linked to an account. It contains the amount of debt referred to the collection agency. It is the creation of this record that, in turn, triggers the interface of information to the collection agency. The method used to interface the information to the agency is defined on the collection agency's record. Refer to Setting Up Collection Agencies for more information.
- If the collection agency is successful in obtaining the funds, simply add a payment. If the payment causes the SA's balance to become zero, the system will automatically close the service agreement. When the system closes a service agreement, it is "removed" from the write-off process. When a write-off process no longer contains active service agreements, the system cancels the write-off process. When a write off process is cancelled, all collection agency referrals are automatically cancelled.
- Collection agency referrals get cancelled by the creation of a new collection agency referral history record (with a type of cancel). This record will be interfaced to the agency in the same manner used to interface a new referral (see above).
- If the collection agency is not successful in obtaining your funds after a given amount of time, you probably want to cancel the referral and write-off the debt. The cancellation of the referral will happen automatically if you design your write-off process to generate a collection agency cancellation X days after the referral. Refer to Setting Up Write Off Process Templates for how to do this. You can cancel a referral manually by simply creating a new collection agency referral history record (with a type of cancel).

  ● **FASTPATH:**
  When you enable the Control Central alert algorithm, C1-COLL-REF, an alert displays when an account has an active collection agency referral. This algorithm is plugged-in on the installation record.

The Big Picture Of Write-off Events

This section describes the various types of write-off events and their lifecycle.

How Are Write-off Events Created?

Write-off events may be created as follows:

- The Write-Off Monitor creates a write-off process when an account has unpaid, final billed service agreements. The write-off process has one or more write-off event(s). Refer to How Does The Write-Off Monitor Work? for more information about how the system creates write-off processes and their events.
- Write-off events are created when an operator creates an ad hoc write-off process. The number and type of events is controlled by the write-off process template defined when the write-off process is created.
- An ad hoc write-off event may be created and linked to an existing write-off process by an operator at their discretion.

  ➤ **NOTE:**
  Bottom line. Most write-off events are created by the system when it creates a write-off process for unpaid, finaled service agreements. If you need to create an ad hoc write-off event, you can either create a write-off

process using a template that contains the desired event OR link the desired event to an existing write-off process.

FASTPATH:
For more information about creating ad hoc write-off processes and events, refer to How To Perform Common Write-off Maintenance Functions.

Types Of Write-off Events

The following table describes the various types of write-off events and what happens when they are completed:

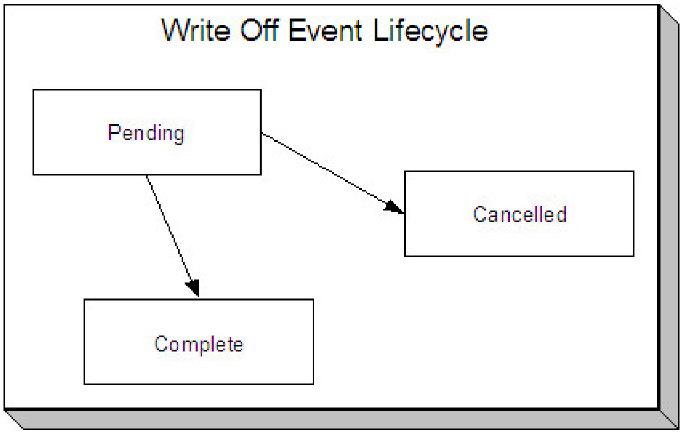
Type Of Write-off Event	What Happens In The System
Affect Credit Rating/Cash-Only	An account credit rating demerit record is created. The number of demerits is defined on the write-off event type.
Cancel Agency Referral	All collection agency referrals associated with the write-off process will be cancelled.
Refer to Agency	The debt associated with the SAs linked to the write-off process will be referred to a collection agency.
Send Letter	A customer contact is created for every financially responsible person linked to the account. The customer contact causes a letter to be produced. The type of letter is defined on the customer contact type control table. The recipient of each letter is defined on Account / Person (those persons marked as **Receiving Notifications**).
Send To Do	A To Do entry is created. Refer to The Big Picture of To Do Entries for more information about To Do entries.
Write Off using Distrib Code	This type of event is used to write-off bad debt in accordance with the distribution codes associated with the financial transactions that caused the debt in the first place. You'd use this method for example if you want to write-off revenue differently than you write-off liabilities. The system accumulates the distribution codes from GL details associated with recent financial transactions linked to each write-off service agreement. When the system has accumulated enough distribution codes (i.e., where the amount associated with the distribution code equals or exceeds the amount to write off), the debt will be transferred to a new or existing write-off service agreement. The type of service agreements to which the debt is transferred is defined on the distribution codes.
Write Off using SA Type	The service agreements linked to the process will be written-off by transferring their debt to a new or existing write-off service agreement. Note: the SA type of the write-off service agreement is defined on the algorithm defined on events of this type.

Generic Algorithm	The system calls the algorithm defined on the write-off event type. This type of event is used when what you need to do isn't handled by one of the above event types.

- **FASTPATH:**
  Refer to Setting Up Write Off Event Types for more information.

Write-off Event Lifecycle

The following diagram shows the possible lifecycle of a write-off event:



Write-off events are initially created in the pending state.

When the system sees a pending event with a trigger date on or before the current date, the system executes the event's activity and completes the event.

- **FASTPATH:**
  For more information about a write-off event's trigger date, see Write-off Event Trigger Date.

A pending event will be cancelled automatically by the system when the service agreements linked to the process are all closed (i.e., they no longer have debt - either because it was paid or transferred to a write-off service agreement). A pending event may be cancelled by an operator at their discretion.

Write-off Event Trigger Date

When a write-off event is created by the system, its trigger date is set in accordance with your date arithmetic preferences. Refer to Calendar vs. Work Days for more information.

How Are Write-off Events Completed

A background process runs periodically (at least daily) that looks for write-off events with a trigger date on or before the current date. For each triggered event, the system executes its activity and then completes it. Refer to Write Off Event Activator for more information.

The Last Write-off Event Should Write Off All Debt Associated With All SAs

The last write-off event will typically transfer all debt from the service agreement's linked to the process to a write off service agreement (linked to the account). This will only happen if you set up the write-off process template accordingly (i.e., the last event in the write-off process template is the kind that writes off debt for every service agreement linked to the write-off process).

How Are Write-off Events Canceled?

The system removes a service agreement from a write-off process when its status becomes closed (i.e., when its balance is zero). Be aware that any type of financial event could cause an SA's balance to fall to zero (e.g., the creation of an adjustment, the application or cancellation of a payment, the cancellation of a bill, ...). When all service agreements are removed, the system cancels all pending write-off events and deactivates the write-off process.

> NOTE:
> **Real time cancellation.** Unlike collection processes, the system cancels write-off processes real time when the service agreement becomes closed (i.e., there is no background process that monitors write-off processes).

Besides the automated cancellation process, an operator may cancel a write-off event at will.

> NOTE:
> **Removing closed service agreements from a write-off process.** Service agreements aren't actually removed from the process. Rather, they are inactivated so a proper audit exists.

Calendar vs. Work Days

When you set up your collection, severance and write-off process templates, you supply information that controls how the system determines the trigger date of each event in the related process. There are two different mechanisms for doing this:

- When you set up your severance process templates, you must define the number of days between each event. For example, the second event (send cutoff warning) may need to be triggered 7 days after the first event (send reminder letter).

- When you set up your collection and write-off process templates, you must define the number of days after the start of the process when each event should be triggered. For example, the second event (send cutoff warning) may need to be triggered 7 days after the start of the collection process.

The system uses this information in conjunction with the account's division's work calendar when it allocates a trigger date to the various collection, severance, and write-off events in your processes. The system offers you the following choices in respect of how it calculates an event's trigger date:

- You can indicate that the trigger date should be set to the next possible workday. For example, if you indicate that the second event is triggered 7 days after the first event, the system will add 7 days to the first event's completion date. It then checks if this is a workday (and not a holiday), if so, this is the trigger date of the event; if not, it assigns the trigger date to the next workday.
- You can indicate that the trigger date should be calculated by counting workdays. For example, if you indicate that the second event is triggered 7 days after the first event, the system will count 7 workdays (using the account's division's work calendar), and set the trigger date accordingly.

You must define which of the above methods is used in the following processes:

- Account Debt Monitor (ADM, ADM2, and C1-ADM). Refer to The Account Debt Monitor for more information.
- Collection Event Trigger (CET and C1-CET). Refer to The Collection Event Activator for more information.
- Severance Event Set Trigger Date (SED and C1-SED). Refer to Set Trigger Date for more information.
- Write-off Monitor (WPM and C1-WPM). Refer to The Write Off Monitor for more information.

The Big Picture Of Payment Arrangements and Pay Plans

The topics in this section describe two different mechanisms that allow a customer to payoff overdue debt in installments.

The Big Picture Of Pay Arrangements

A payment arrangement is an agreement with a customer to payoff severely overdue debt in **billed** installments. Bills sent to customers with payment arrangements contain charges for both their current services and their payment arrangement installment amount.

> **NOTE:**
> **Nomenclature.** Some people refer to payment arrangements as "current bill plus" agreements because the customer's bills contain charges for both their current debt plus their installment amount. After the customer has paid off their overdue debt, the customer's bill only contains charges for their current debt.

The topics in this section describe how to set up a payment arrangement and how the system monitors the ongoing arrangements.

Creating Payment Arrangements

When you create a payment arrangement, you are actually creating a service agreement. This service agreement is just like other service agreements in that:

- It holds debt.

- It is periodically billed.
- When a payment is received, the service agreement's debt is reduced.
- If the service agreement becomes delinquent, a collection process is initiated to collect the overdue debt.

Debt is transferred to a payment arrangement service agreement (PA SA) from the customer's delinquent service agreements (SAs) at the inception of the payment arrangement.

When you transfer delinquent debt from the delinquent SAs to the PA SA, the debt is removed from the delinquent SAs. If you transfer all debt from the delinquent SAs, the customer will no longer be in arrears in a given debt class (and if the customer is no longer in arrears, active collection and severance processes will be cancelled).

NOTE:
Use the Payment Arrangement Transaction. You could do the above functions by adding a new service agreement and creating transfer adjustments. However, this is tedious. Rather, use the Payment Arrangement transaction. This transaction creates the PA SA, transfers debt to it, and sets up the installment amount. This transaction is also used if you need to break or cancel the payment arrangement.

Installment, Payoff and Current Amounts

WARNING:
If you do not understand the difference between payoff balance and current balance, refer to Current Amount versus Payoff Amount.

When you set up a payment arrangement service agreement (PA SA), you transfer delinquent debt to the PA SA using transfer adjustments. After moneys are transferred, the system sets the PA SA's current balance to zero. At this point, neither the original service agreements nor the PA SA have delinquent debt. If the customer neglects to pay their payment arrangement, the PA SA will fall into arrears and a collection process will ensue. If the customer neglects to pay their previously delinquent SAs, they will again fall into arrears and a collection process will ensue.

PA SAs start their life with a non-zero payoff balance (i.e., they have debt when first started). This debt is transferred from the normal service agreement(s) whose outstanding debt necessitated the creation of the PA SA.

The installment amount that the customer is billed is determined by the number of installments used to payoff the debt. For example, if the customer owes $500 on their electric and water service agreements and they want to pay this off in 10 installments, you'd set up the installment amount to be $50. The installment amount is saved on the PA SA's recurring charge amount. If the customer again falls into arrears on their normal service agreements, you can transfer additional delinquent debt to the PA SA. You can also change the installment amount as needed.

A PA SA's payoff balance typically differs from its current balance. The payoff balance is the amount of debt remaining to be paid off under the terms of the payment arrangement. The current balance is the installment amount that has been billed but not paid. For example, a customer who is paying off $500 with 10 installments of $50 would have an initial payoff balance of $500 and a current balance of $0. After the first bill, the PA SA would still have a payoff balance of $500, but its current balance would be $50. When the customer pays, the PA SA's payoff balance would fall to $450 and its current balance would return to $0.

The following table contains a financial example of a customer who sets up a payment arrangement to payoff $1,000 of debt in $10 installments.

Event	Normal SA's GL Accounting	PA SA's GL Accounting	Normal SA's Current Balance	Normal SA's Payoff Balance	PA SA's Current Balance	PA SA's Payoff Balance

Prior to creation of payment arrangement	N/A	N/A	1000	1000	N/A	N/A
Transfer debt from normal SA(s) to PA SA	Xfer 1000 A/R <1000>	PA A/R 1000 Xfer <1000>	0	0	1000	1000
Set current balance to zero on PA SA	N/A	N/A	0	0	0	1000
Customer is billed ($50 for new debt and $10 of payment arrangement debt)	A/R 50 Revenue <50>	N/A	50	50	10	1000
Customer pays $60	Cash 50 A/R <50>	Cash 10 PA A/R <10>	0	0	0	990

When the customer pays off the payment arrangement debt, the system automatically closes the PA SA after it final bills (assuming the PA SA's SA type references a bill segment type that has a bill segment creation algorithm of Recurring Charge With Auto Stop).

Monitoring Payment Arrangements

The PA SA should belong to its own debt class (let's call it Payment Arrangement Debt) so that you can have stricter collection criteria for payment arrangement debt (as compared to normal SAs). Because there will be a new debt class, there will be a unique collection class control (CCC) for payment arrangements. This CCC will have debt criteria associated with payment arrangement debt. If these criteria are violated, we will kick off a collection process that should have 1 collection event - Start Severance.

The severance process template for PA SAs will have 1 severance event that calls the Break Payment Arrangement Event algorithm. This algorithm does the following:

- Cancels ALL adjustments that were used to transfer the debt to the payment arrangement (identified by the XFER adjustment type on the PA SA's SA type). When these are cancelled, the original arrearage will be reinstated under the original SAs - this debt should be rather old by this point.
- Syncs up current balance with payoff balance on the PA SA.
- Makes the PA SA pending stop (SA activation will stop the SA when it next runs).
- If there is a credit left on the PA SA (because payments were made against the arrangement), the credit will be distributed amongst the account's debt using the standard distribution algorithm. Because the payment arrangement debt that was reinstated should be rather old, it should get relieved first. This relief will occur via transfer adjustments from the PA SA to the original SAs.
- If there is a debit left (e.g., because LPC were issued or some other type of adjustment was created by an operator), the debt will be transferred back to one of the SAs from which the arrangement was originally created.
- Inserts a characteristic under the PA SA to indicate that it has been broken (we need this for the account debt monitor (ADM) a few steps down).
- Inserts a row on the account debt monitor trigger. This trigger will cause the account to be reviewed by the ADM the next time it runs.

➤ **NOTE:**

The PA SA must final bill before it closes. The PA SA will only close after the PA SA is final billed. This is OK as it won't have any money left on it.

When the ADM next runs, it will analyze the account's reinstated debt. We recommend creating a new override collection criteria for the normal debt class that will return a value of true if the account has a closed payment arrangement that has been broken in the last X days (where X is a parameter of the override collection criteria's algorithm). If this algorithm returns a true, kick off a unique collection process template (that has nasty events). A sample algorithm of this type is supplied in the base package - COLL COND PA.

To complete this discussion, we have to worry about the situation when the final bill of a payment arrangement goes unpaid. In this situation, the payment arrangement is stopped and will therefore not be processed by the ADM. In this case, the write off monitor will process the PA SA after its final bill's due date and a write-off process will start. This write off process will have a single event that calls the Break Payment Arrangement algorithm (described above). After the FT's are issued in this event, the SA will close (because it's been final billed and its balance will go to zero).

The Big Picture Of Pay Plans

A pay plan (PP) is an agreement with a customer to make payments on specific dates. Pay plans differ from payment arrangements in that pay plans have user-defined scheduled payment dates, which are independent from the customer's billing dates. In other words, payment arrangements appear on the customer's bills, pay plan scheduled payments do not.

If a customer is in arrears and you want to receive payments on specific dates (as opposed to with the customer's regular bills), you would set up a pay plan and define the dates on which you expect the payments.

The topics in this section describe how pay plans work.

A Pay Plan Has One Or More Scheduled Payments

When you create a pay plan for an account, you must define the number of scheduled payments and their respective amounts. There is no limit to the number of scheduled payments that may be set up under a pay plan.

Automatic Payments Can Be Created On The Scheduled Payment Dates

The system will create automatic payments on a pay plan's scheduled payment dates if:

- The account is set up for automatic payment (as described under How To Set Up A Customer To Pay Automatically), and
- The payment method defined on the pay plan indicates automatic payment is being used

The background process called PPAPAY is responsible for creating these automatic payments. It does this by calling the automatic payment creation algorithm plugged in on the installation record.

➤ **NOTE:**

If the **Autopay Creation Option** on the installation record is set to Create On Extract Date, the automatic payment is NOT distributed and frozen when the automatic payment is initially created. Rather, a separate background process (APAYDSFR) distributes and freezes the automatic payment on the automatic payment

GL distribution date (refer to Automatic Payment Dates for more information on how this date is calculated). Refer to Automatic Payments for more information.

A Pay Plan Insulates Overdue Debt From The Account Debt Monitor

A pay plan's scheduled payments are used by the account debt monitor as "pseudo payments" that relieve the account's debt before it is subjected to the collection criteria (refer to How Does The Account Debt Monitor Work for more information about collection criteria).

It's important to understand that a pay plan only insulates the account's debt that belongs to the pay plan's debt class. Therefore, if a customer has debt that belongs to two debt classes (e.g., normal debt and 3 rd party pass through debt), you would need to set up a separate pay plan for each debt class (assuming both types of debt are covered by a pay plan). Refer to Different Collection Criteria For Different Customers and Different Debt for more information about debt classes.

A Pay Plan Must Reference A Pay Plan Type

When you create a pay plan, you must define its pay plan type. The pay plan type controls the following functions:

- The debt class whose debt is insulated by the pay plan.
- The type of algorithm (if any) that is executed when the pay plan is broken. You might use such an algorithm to affect the customer's credit rating when the pay plan is broken.

A Pay Plan May Reference A Third-Party Payor

In addition to referencing the account whose debt is insulated by the pay plan, the pay plan must also reference the account that is responsible for making the payments. We refer to this second account as the pay plan's "payor".

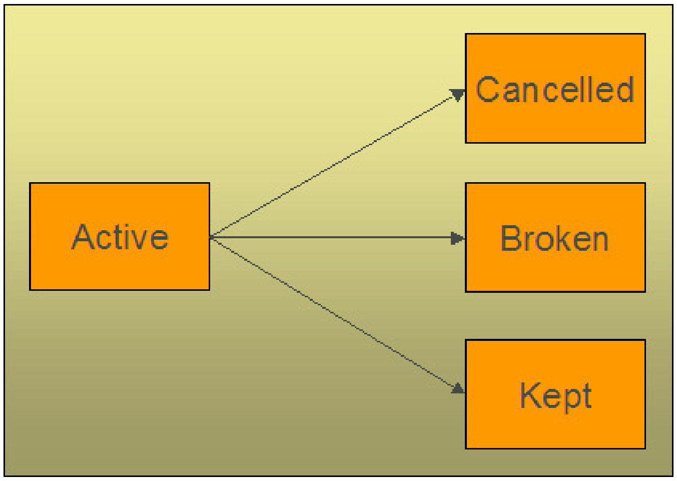
While the payor's account is typically the same as the account whose debt is insulated by the pay plan, you can indicate a third-party payor (e.g., a social service agency) is responsible for making the pay plan's scheduled payments.

If your organization allows third-party payors, you can define each on the third-party payor control table. This control table exists to simplify the data-entry effort when you create a pay plan (as it defines the account associated with the third-party payor).

➤ **NOTE:**
If a pay plan does not reference a third-party payor, any non-third-party payor (i.e., any account that is not defined in the third-party payor control table) can make payments on behalf of the customer. If a pay plan references a third-party payor, only payments made by the third party on behalf of the customer are counted towards the fulfillment of the pay plan.

The Lifecycle Of A Pay Plan

The following diagram shows the possible lifecycle of a pay plan:



The following points explain this lifecycle:

- Pay plans are initially created in the active state. Active pay plans are monitored for compliance by The Pay Plan Monitor.
- A pay plan may be cancelled as follows:
    - A user can cancel a pay plan at will.
    - **When a SA is stopped AND there are no other active SAs in the same debt class, all** active **pay plans associated with the account and debt class will be will be** cancelled**.**
    - The activation of a collection event that calls the "cancel pay plan" algorithm will cancel all active pay plans associated with the collection process's debt class. You may want to use such a collection event if your organization cancels active pay plans when new debt causes a collection process to kick-off. Note, the base package algorithm that performs this function will not cancel the pay plan if it's associated with a 3rd party payor.
- The Pay Plan Monitor causes active pay plans to become broken if sufficient payments have not been made to satisfy the pay plan's scheduled payments.
- The Pay Plan Monitor causes active pay plans to become kept when it detects that sufficient payments have been made to satisfy the pay plan's scheduled payments.

Highlighting The Existence Of Broken / Kept / Active and Denied Pay Plans

You can define on the installation record plug-in algorithms that format alert messages. (Refer to Installation Options - Algorithms for additional information.) We recommend that you take advantage of the following algorithms to highlight pay plans:

- Highlight pay plans in a given status. This algorithm is used to highlight pay plans in a given state (broken, kept, cancelled) that were started within the last X days.
- Highlight customer contacts of a given type. This algorithm would be used to highlight customer contacts of a given type that were created within the last X days. This would be useful if you create a specific type of customer contact when you deny a pay plan. Some utilities do this to prevent customers from shopping around.

In addition, you can define account-specific alerts to highlight customers that should never be allowed to have a pay plan (for whatever reason).

A Pay Plan Must Reference A Payment Method

When you create a pay plan, you must define how the customer will make the payments be referencing a payment method. Examples of payment methods include: In Person , Wire Transfer, By Post, Express Mail, etc.

The payment method is more than just documentation as it defines the number of grace days the customer has to make the pay plan's scheduled payments. For example, if you set up the payment method control table to indicate that payments made By Post have 3 grace days, then the customer has up to 3 days after each scheduled payment date to make the payment. If payment is not received by the scheduled payment date plus the grace days, the pay plan will be marked as broken (and the ADM will be triggered).

The Pay Plan Monitor

●

FASTPATH:
Please understand the concepts described in The Lifecycle Of A Pay Plan and The Tendering Account May Differ From The Account Whose Debt Is Relieved before reading this section.

The Pay Plan Monitor background process (referred to as PPM) is responsible for monitoring active payment plans. This process can cause a pay plan (PP) to become kept or broken (or being left as active).

➤

NOTE:
When is a pay plan marked as broken / kept? It's important to understand that only the PPM can cause a pay plan to become kept or broken. This means that if a customer makes a payment that satisfies a pay plan, the pay plan will only be marked as kept when the pay plan monitor next runs. Analogously, if a payment is cancelled, nothing will happen to an active pay plan until the PPM next runs. When the PPM next runs, it will see that the scheduled payment was not kept and it will break the pay plan and schedule the ADM to be executed. When the ADM next executes, it will create a collection process (because the customer's debt will no longer be insulated by the pay plan's scheduled payments).

NSF Cancellations After A Pay Plan Is Kept. If a payment is cancelled due to non-sufficient funds (NSF) after a pay plan is marked as kept, the pay plan will remain kept. But keep in mind that the pay plan's account is scheduled for review by the ADM when a payment is cancelled due to NSF. When the ADM reviews the account's debt, it will no longer have an active pay plan to insulate it and the account's debt will likely trigger a new collection process. Refer to How Pay Plans Affect The Account Debt Monitor for more information.

The following points describe, at a high level, how the PPM monitors a pay plan (PP) for compliance.

- The system selects all frozen, non-cancelled payment segments associated with the PP's account and debt class where:
  - The payment date is after the start date of the pay plan, and
  - The payment's pay event has at least one tender that references the pay plan's payor.
- The system logically reduces / removes past and current scheduled payments (starting with the earliest scheduled payment) until the total amount of payment segments is exhausted (or there are no more historical / current scheduled payments).

➤

NOTE:

Paying pay plans in advance. Scheduled payments with a future date are not logically removed / reduced. This means that if a customer makes advance payments on a pay plan, it will not be marked as kept until all scheduled payment dates are in the past.

- If all scheduled payments have been logically removed, the pay plan is marked as kept.
- If there exist scheduled payments where the pay date + grace days (grace days are defined on the pay plan's payment method) is before the current date (i.e., a payment doesn't exist for a scheduled payment):
  - The pay plan is marked as broken.
  - The PP's break algorithm (if any) is called (note, for European / Australian pay plans, there are scenarios where the break algorithm can cause the pay plan to become unbroken - when there aren't at least two missed, historical scheduled payments).
  - An ADM trigger is stored for the PP's account. This will cause the account to be reviewed by the ADM the next time it runs. And because the pay plan is broken, its scheduled payments will no longer insulate the account's arrearage.

⚠ **IMPORTANT:**
It's important that you schedule the PPM to run before the ADM so that it can break unpaid payment plans prior to the ADM subjecting the account's debt to collection criteria. Refer to How Pay Plans Affect The ADM for more information.

How Pay Plans Affect The Account Debt Monitor

As described under A Pay Plan Insulates Overdue Debt From The Account Debt Monitor, a pay plan's scheduled payments insulate an account's debt from the ADM. This section describes how this is accomplished.

WARNING:
You should understand the concepts in How Does The Account Debt Monitor Work and The Tendering Account May Differ From The Account Whose Debt Is Relieved before reading the following.

➤ **NOTE:**
The ADM will be triggered when a pay plan is broken. Refer to The Pay Plan Monitor for an explanation of how the ADM is triggered when a pay plan is broken.

Before the account debt monitor background processes (ADM, ADM2, or C1-ADM) subjects an account's debt to the collection criteria, it calls the debt's debt class's Override Arrears Algorithm (this is an optional plug-in spot on Debt Class). This algorithm is passed the debt class's aged debt and manipulates it as follows:

- First, a list of all past, present and future scheduled payments associated with the account and debt class's active pay plans is constructed.
  - If multiple payors are encountered (because the customer has multiple pay plans and these have different payors), a separate list of scheduled payments is maintained for each payor.
- Next, for each payor, retrieve the total amount of frozen, non-cancelled payment segments made on behalf of the pay plan's account and debt class.

  - Select all frozen, non-cancelled payment segments associated with the pay plan's account and debt class whose pay date is >= pay plan's start date and the pay segment's event has at least one tendering account associated with the pay plan's payor.
- Next, logically reduce / remove past and current scheduled payments (starting with the earliest scheduled payment) until the payor's payment amount is exhausted (or there are no more historical / current scheduled payments). Future scheduled payments cannot be remove / reduced.
- Finally, reduce the passed in aged debt with any unpaid scheduled payments.

NOTE:
This logic is not "hard coded". Rather, the mechanism used to use a pay plan's scheduled payments to reduce debt is defined in an algorithm defined on the pay plan's debt class. The contents in this section describe how a base package algorithm works. Because it's an algorithm, you can introduce whatever logic you please.

The following is an example of how pay plans affect aged debt.

Date	Event	SA's Arrears	SA's Balances	Scheduled Payments
Prior to creation of the PP 1/18/2000		$1,000 - 90 days old $1,600 - 60 days old $1,900 - 30 days old	Current: $4,500 Payoff: $4,500	
1/18/2000	Pay Plan created. The $4,500 in future scheduled payments offsets the existing $4,500 of aged debt.	$1,000 - 90 days old $1,600 - 60 days old $1,900 - 30 days old **De facto ADM debt: $0**	Current: $4,500 Payoff: $4,500	1/20/2001 $1,500 2/01/2001 $1,500 2/07/2001 $1,500
1/20/2001	The customer pays $1,500. There exists $3,000 of future scheduled payments that offset the arrears	$1,100 - 62 days old $1,900 - 32 days old **De facto ADM debt: $0**	Current: $3,000 Payoff: $3,000	1/20/2001 $1,500 "Paid" 2/01/2001 $1,500 Future 2/07/2001 $1,500 Future
1/20/2001	ADM runs. The $3000 in future scheduled payments offsets the Current Balance of $3000, so CC events not created.	$1,100 - 62 days old $1,900 - 32 days old **De facto ADM debt: $0**	Current: $3,000 Payoff: $3,000	1/20/2001 $1,500 "Paid" 2/01/2001 $1,500 Future 2/07/2001 $1,500 Future
1/24/2001	A new bill is created for $400	$1,100 - 62 days old $1,900 - 32 days old $400 - 1 day old **De facto ADM debt: $400 - 1 day old**	Current: $3,400 Payoff: $3,400	1/20/2001 $1,500 "Paid" 2/01/2001 $1,500 Future 2/07/2001 $1,500 Future
2/2/2001	Pay Plan Monitor runs. PP marked as Broken because the 2/1/2001 scheduled payment has not been paid (assuming no grace period on the	$1,100 - 74 days old $1,900 - 44 days old $400 - 8 days old	Current: $3,400 Payoff: $3,400	1/20/2001 $1,500 "Paid" **2/01/2001 $1,500 "Late"** 2/07/2001 $1,500 Future

	pay plan's payment method)			
2/2/2001	ADM runs. There are no active pay plans and therefore there is nothing to insulate the customer's debt. Therefore the aged debt will be subjected to the collection criteria and an appropriate collection process will be created.	$1,100 - 74 days old $1,900 - 44 days old $400 - 8 days old **De facto ADM debt is the same as above (i.e., rather old)**	Current: $3,400 Payoff: $3,400	**Pay plan is broken and therefore its scheduled payments cannot be used.**

Collection Process / Severance Process Cancellation

When a pay plan (PP) is created, the system determines if it can cancel active collection and severance processes associated with the pay plan's account and debt class. It does this because a pay plan's scheduled payments act as "pseudo payments" that relieve the account's debt (temporarily). The following points describe how this works:

- The system attempts to cancel collection processes by calling the Collection Process Cancel Criteria Algorithm defined on the debt class that is associated with the collection process. This algorithm is meant to cancel a collection process if the sum of ALL service agreements in the debt class have debt less than a given threshold amount. Because of the existence of the pay plan, the actual debt will be temporarily reduced by the amount of the pay plan's scheduled payments before it is compared to the threshold amount (see How Pay Plans Affect The Account Debt Monitor for more information about how debt is reduced). Note: this temporary reduction will only occur if you have plugged in the appropriate pay plan debt reduction algorithm on the debt class.
- It attempts to cancel severance processes by calling the Severance Process Cancel Criteria Algorithm defined on the debt class that is associated with the severance process. This algorithm is meant to cancel a severance process if the sum of ALL service agreements in the debt class have debt less than a given threshold amount. Because of the existence of the pay plan, the actual debt will be temporarily reduced by the amount of the pay plan's scheduled payments before it is compared to the threshold amount (see How Pay Plans Affect The Account Debt Monitor for more information about how debt is reduced). Note: this temporary reduction will only occur if you have plugged in the appropriate pay plan debt reduction algorithm on the debt class.

If collection / severance processes still exist for the account / debt class associated with the pay plan, a warning is issued.

Interesting Pay Plan Facts

The following points describe a variety of interesting facts about pay plans (PP):

- An account may have many active pay plans. However, only 1 pay plan may be active for a given account / debt class / payor at any point in time.
- The existence of a pay plan has no impact on payment distribution.
- When a SA is stopped, if the SA's debt class has an active PP AND there are no other active SAs in the same debt class, the PP will be cancelled
  - The cancel reason will be "cancelled by system" (as opposed to "cancelled by user")
- If necessary, different collection / severance processes can be triggered if a broken PP is detected (via the override algorithms on CCC and write-off control - we do NOT provide such algorithms).

Setting Up Pay Plan Control Tables

This section describes the control tables needed to set up pay plans.

Setting Up Pay Plan Types

Pay plan types control what is done by a given pay plan. Open **Admin** > **Credit & Collection** > **Pay Plan Type** to define your pay plan types.

- **FASTPATH:**
  For more information refer to The Big Picture Of Pay Plans for more information.

Description of Page

To modify a pay plan type, simply move to a field and change its value. To add a new pay plan type, press + to insert a row, then fill in the information for each field. The following fields display:

Pay Plan Type The name of the pay plan type.

Description A meaningful description of the pay plan type.

Broken Algorithm This algorithm is called when a pay plan is broken. Refer to The Pay Plan Monitor for more information about how pay plans are broken.

If you haven't done so already, you must set up this algorithm in the system. To do this, create a new algorithm (refer to Setting Up Algorithms). On this algorithm, reference an Algorithm Type that breaks a pay plan. Click here to see the algorithm types available for this plug-in spot.

Debt Class The debt class covered by pay plans of this type. Refer to Setting Up Debt Classes for more information.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference CI_PP_TYPE.

Setting Up Payment Methods

Payment methods are used to describe how a customer intends to make their pay plan's scheduled payments. Open **Admin** > **Credit & Collection** > **Pay Method** to define your payment methods.

- **FASTPATH:**
  For more information refer to A Pay Plan Must Reference A Payment Method for more information.

Description of Page

To modify a pay method, simply move to a field and change its value. To add a new pay method, press + to insert a row, then fill in the information for each field. The following fields display:

Pay Method The name of the payment method.

Description A meaningful description of the payment method.

Grace Days The number of days added to the scheduled payment date. The ADM will consider the pay plan to be broken if payment is not made by the scheduled date plus the grace days.

Auto Pay If the pay method is marked as being for **Auto Pay**, the PPAPAY background process will automatically create an automatic payment on the pay plan's scheduled payment dates IF the account has been set up for automatic payment.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference CI_PAY_METH.

Setting Up Third Party Payors

Pay plans support optional third-party payors. Open **Admin** > **Credit & Collection** > **Third Party Payor** to define your third-party payors.

> **NOTE:**
> A third-party payor refers to an account. You must set up the account before you can create a third-party payor.

> **FASTPATH:**
> Refer to A Pay Plan May Reference A 3rd Party Payor for more information.

Description of Page

The following fields display for each third party payor:

Third Party Payor Provide a meaning id for the third-party payor that can be easily recognized when setting up a pay plan.

Description A meaningful description of the payor.

Account ID The account that is used for this payor. It is this account that is tracked as the "payee" of any payments made towards a third party payor's pay plan.

Active Check this box if the payor is currently available to participate in pay plans.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference CI_THRD_PTY.

Creating Collection, Severance & Write-Off Procedures

Your collection procedures define how your organization collects overdue debt. Your severance procedures define how your organization severs service agreements when collection attempts fail. Your write-off procedures define how your organization writes off finaled debt. In this section, we describe how to set up the data that controls these procedures.

> **FASTPATH:**

For more information about collection, severance and write-off procedures, see The Big Picture Of Credit & Collections (C&C).

WARNING:
There are innumerable ways to design your collection, severance and write-off procedures. Some designs will result in easy long-term maintenance, others will result in maintenance headaches. In this section, we provide information to help you understand the ramifications of the various options. Before you set up your production collection, severance and write-off procedures, we encourage you to gain an intuitive understanding of these options by using the system to prototype the alternatives.

Designing Your Collection Procedures

The design of your collection procedures is an iterative process. Over time, you will develop intuitive skills that will allow you to skip some iterations. However, when you're starting out, we recommend you use the following matrix as your guide. When the matrix is complete, you're ready to set up the collection process control tables.

SA's Debt Class	Account's Collection Class:	Account's Collection Class:

The topics discussed below will gradually complete this matrix using a simple case study.

FASTPATH:
For more information about how the information in this matrix is used to monitor your customers' debt, refer to Different Collection Criteria For Different Customers And Different Debt.

Designing Your Debt Classes

Multiple debt classes are needed when you have different collection procedures for different types of service agreements. If all service agreement debt is collected the same way, then you'll have just one debt class (call it Generic). However, if you're like many organizations, you will have multiple debt classes. The following points will help you understand why:

- If you have both regulated and unregulated service, you probably have different debt classes for each type of service. Why? Because your local regulators control how you collect and cutoff overdue, regulated debt. For unregulated debt, your organization controls how overdue debt is collected.
- If your customers make charitable donations, you will have a charitable contribution debt class. Why? Because you probably send a different type of letter when the customer falls into arrears on their charitable contributions. You also can't cut off their water service if they don't make their charitable contributions.
- If you levy deposits, you will probably have a deposit debt class. Why? Because you probably respond differently if the customer doesn't pay their deposit (e.g., you may decide to cut off their electric service until the deposit is paid).
- If you allow customers to make payments on Non-Billed Budgets, you will probably have a Non-Billed Budget debt class. Why? Because you probably respond differently if the customer doesn't pay their Non-Billed Budget (e.g., you may decide to expire the Non-Billed Budget but not affect their other service since the Non-Billed Budget is a way for customers to prepay for upcoming bills).

- If you write-off uncollectable debt, you will need another debt class for write-off service agreements. Why? Because when you write-off debt in the system, you transfer the uncollectable debt from the original service agreement(s) to a write-off service agreement. The write-off service agreement holds this debt forever (or until it is paid). You need to use a different debt class for the write-off service agreements because they aren't subject to collection criteria.
- If you use the system to charge your organization's company usage, you'll need another debt class (we refer to it as the "N/A" debt class below). Why? Because all service agreements must have a debt class, even those that will never have debt.

SA's Debt Class	Account's Collection Class	Account's Collection Class
Regulated		
Unregulated		
Charitable Contribution		
Deposit		
Non-Billed Budget		
Write Off		
N/A		

Designing Your Collection Classes

Multiple collection classes are needed when any debt class has different collection rules depending on the type of customer. If all customers within all debt classes are collected the same way, then you'll just have a single collection class (call it Generic). However, if you're like many organizations, you will have multiple collection classes.

Consider unregulated debt. For commercial/industrial customers, you probably don't worry until they owe you more than, say, $100 after 20 days. For residential customers, you probably don't worry until they owe you more than, say, $5 after 20 days. In this situation, you will have at least two collection classes: one for large customers, the other for residential customers.

In our example, we are assuming you have two collection classes: Residential and Commercial/Industrial.

SA's Debt Class	Account's Collection Class:	Account's Collection Class:
	Residential	Commercial/Industrial
Charitable Contribution		
Regulated		
Unregulated		
Write Off		
Company Usage		

Designing Collection Class Controls

At this point we have the rows and columns defined in our matrix. Now it's time to work on the individual cells.

Each cell should have a "collection class control" that defines its collection criteria and what to do if the criteria are violated. If a cell doesn't have a collection class control, this means you don't have any debt associated with that combination of collection class and debt class. So, we'll mark each cell without debt with "N/A".

SA's Debt Class	**Account's Collection Class:** **Residential**	**Account's Collection Class:** **Commercial/Industrial**
Regulated		N/A
Unregulated		
Charitable Contribution		N/A
Deposit		
Write Off		
N/A		

Next, we'll mark each cell for debt classes whose debt isn't collectable (i.e., the write-off and N/A debt classes).

SA's Debt Class	**Account's Collection Class:** **Residential**	**Account's Collection Class:** **Commercial/Industrial**
Regulated		N/A
Unregulated		
Charitable Contribution		N/A
Deposit		
Write Off	N/A	N/A
N/A	N/A	N/A

NOTE:
If the Account Debt Monitor encounters debt associated with a non-existent collection class control, it will issue an error.

Determining the collection criteria in each remaining cell can be straightforward or complicated; it depends on how your organization works. Our case study assumes the following:

- For charitable debt, if the customer is more than $0 in arrears by more than 20 days, kick off the "charity reminder" collection process. We'll talk more about this collection process later.
- For regulated / residential debt, if the customer is more than $15 in arrears by more than 20 days, kick off the Normal Regulated collection process. We'll talk more about this collection process later.
- For unregulated / residential debt, if the customer is more than $5 in arrears by more than 20 days, kick off the Normal Unregulated collection process. We'll talk more about this collection process later.
- For unregulated / commercial-industrial debt we have multiple criteria:
  - Highest priority. If the customer is more than $10,000 in arrears by more than 20 days, kick off the Large Overdue Debt collection process. We'll talk more about this collection process later.
  - Lower priority. If the customer is more than $100 in arrears by more than 20 days, kick off the Normal Unregulated collection process. We'll talk more about this collection process later.
- For deposit debt (regardless of collection class) we have multiple criteria:
  - Highest priority. If the customer is more than $5 in arrears by more than 50 days, kick off the Deposit Severely Overdue collection process. We'll talk more about this collection process later.

- Lower priority. If the customer is more than $15 in arrears by more than 20 days, kick off the Deposit collection process. We'll talk more about this collection process later.

Given the above, our matrix will look as follows:

SA's Debt Class	Account's Collection Class:	Account's Collection Class:
	Residential	Commercial/Industrial
Charitable Contribution	If > $0 is older than 20 days, start Charity Reminder collection process.	N/A
Regulated	If > $15 is older than 20 days, start Normal Regulated collection process.	N/A
Unregulated	If > $5 is older than 20 days, start Normal Unregulated collection process.	Highest priority: If > $10,000 is older than 20 days, start Large Overdue Debt collection process. Lower priority: If > $100 is older than 20 days, start Normal Unregulated collection process.
Deposit	Highest priority: If > $5 is older than 50 days, start Deposit Severely Overdue collection process. Lower priority: If > $15 is older than 20 days, start Deposit collection process.	Highest priority: If > $5 is older than 50 days, start Deposit Severely Overdue collection process. Lower priority: If > $15 is older than 20 days, start Deposit collection process.
Write Off	N/A	N/A
N/A	N/A	N/A

Designing Your Collection Class Control Overrides

> **WARNING:**
> Your collection needs may not require any overrides for your collection class control matrix and therefore this section may not be relevant.

The following matrix will help you design your collection class overrides. When the matrix is complete, you're ready to set up the collection class control tables.

Notice that the matrix has two dimensions: one is dependent on collection condition algorithms; the other is dependent on the collection class controls designed in the previous section. Collection condition algorithms are confusing. Think of them as optional conditions that, if met, will subject the collection class control's debt to different collection criteria.

Each cell in the matrix contains the collection criteria that will be applied to the account's debt when the collection condition is met (i.e., the same type of criteria - dollars and days and collection process - are defined in each cell).

We label the first collection condition as the Default. The collection criteria associated with this column will be used to analyze an account's debt when none of the other conditions applies. We'll start by indicating the Default collection criteria (this was defined in the previous section).

SA's Debt Class	Account's Collection Class:	Account's Collection Class:
	Default	Credit Rating < Threshold
Residential Charitable Contribution	See default collection criteria defined in previous section.	

Residential Regulated	See default collection criteria defined in previous section.
Residential Unregulated	See default collection criteria defined in previous section.
Commercial-Industrial Unregulated	See default collection criteria defined in previous section.
Residential Deposit	See default collection criteria defined in previous section.
Commercial-Industrial Deposit	See default collection criteria defined in previous section.

If a different collection process OR criteria should be used when other conditions are met, you should indicate such by defining the collection criteria in the cell. For example, if we assume that all unregulated residential debt has a different collection process when the account's credit score is less than the threshold credit rating on the installation record or the account’s CIS Division, our matrix will look as follows:

SA's Debt Class	Account's Collection Class:	Account's Collection Class:
	Default	Credit Rating < Threshold
Residential Charitable Contribution	See default collection criteria defined in previous section.	
Residential Regulated	See default collection criteria defined in previous section.	
Residential Unregulated	See default collection criteria defined in previous section.	Override: If Credit Rating is lower than the credit rating threshold: If > $5 is older than 15 days, start Risky Unregulated collection process.
Commercial-Industrial Unregulated	See default collection criteria defined in previous section.	
Residential Deposit	See default collection criteria defined in previous section.	
Commercial-Industrial Deposit	See default collection criteria defined in previous section.	

Once the matrix is complete, you're ready to design your collection process and collection event types.

NOTE:

The collection conditions are limited by your imagination (and business requirements). We have provided the collection conditions you see above as an example; we don't expect you'll be able to use the exact conditions we supply. Your conditions will be based on any number of factors. For example, if you have different collection criteria that apply during winter months, you should add a new collection condition (called Winter Season). Or if you have different criteria based on years of service, you could have another condition.

New collection conditions may require programming. See How To Add A New Algorithm for more information.

Designing Collection Process Templates & Collection Event Types

The following table shows the collection process templates referenced in the previous section's matrix. Adjacent to each process are its events and an indication of when they are triggered.

Collection Process Template	Collection Event Template	Triggered X Days From Start Of Collection Process
Charitable Contribution Reminder	Charity courtesy reminder letter	0
	Start severance	15
Normal Regulated	Regulated courtesy reminder letter	0
	Regulated second notice letter	10
	Start severance	15
Large Overdue Debt	Large debt courtesy reminder letter	0
	To Do for large overdue debt	3
	Large debt second notice letter	10
	Start severance	15
Normal Unregulated	Unregulated courtesy reminder letter	0
	Unregulated second notice letter	5
	Start severance	10
Risky Customer Unregulated	Unregulated risky letter	0
	Start severance	5
Deposit	Deposit reminder	0
Deposit Severely Overdue	Create To Do entry	0

If we extract each unique event type from the above table, we end up with the following:

Collection Event Type	Event Type
Charity courtesy reminder letter	Send Letter - CHARIT REMIN
Start severance	Start Severance Process
Regulated courtesy reminder letter	Send Letter - REGUL REMIN
Regulated second notice letter	Send Letter - REGUL second
Large debt courtesy reminder	Send Letter - LARGE REMIN
Risky debt courtesy reminder	Send Letter - RISKY REMIN
To Do for large overdue debt	Issue To Do
Large debt second notice letter	Send Letter - LARGE second
Unregulated courtesy reminder letter	Send Letter - UNREG REMIN
Unregulated second notice letter	Send Letter - UNREG second
Deposit reminder	Send Letter - DEPOS REMIN

To Do for deposit severely overdue	Issue To Do

Now you're (almost) ready to set up your collection procedures.

Defining Cancellation Process Auto Cancellation Criteria

The topics in the section How Are Collection Processes Cancelled describe the two algorithms that play a part in the cancellation of a collection process. It also describes when to use what type of algorithm. Please read this section and then set up the appropriate cancellation criteria on your Debt Classes, and optionally, on your Collection Process Templates.

Setting Up Collection Procedures

In the previous section, Designing Your Collection Procedures, we presented a case study that illustrated a mythical organization's collection procedures. In this section, we'll explain how to set up the control tables to implement these procedures:

Setting Up Collection Event Types

Collection event types control what is done by a given collection event. Open **Admin** > **Credit & Collection** > **Collection Event Type** > **Add** to define your collection event types.

Description of Page

Enter a unique **Collection Event Type** and **Description** for the collection event type.

Enter the **Collection Event Type**. Permissible values are: Affect Credit, Rating/Cash-Only, Cancel Budget, Generic Algorithm, Send Letter, Create To Do Entry, Start Severance Process. The following discussion describes the parameters that must be defined for each type of collection event.

The Affect Credit Rating/Cash-Only collection event type causes a credit rating demerit record to be linked to the account. This record is constructed using the following **Parameters**:

- Use **Credit Rating Points** to define this event's affect on the account's credit rating. This should be a negative number. An account's credit rating is equal to the beginning credit rating amount defined on the installation record or the account’s CIS Division plus the sum of credit rating demerits that are currently in effect. When an account's credit rating is less than the credit rating threshold defined on the installation record or the account’s CIS Division, the account's credit rating is displayed as an alert on Control Central.
- Use **Cash-Only Points** to define this event's affect on the account's cash-only score. This should be a positive number. When an account's cash-only score exceeds the cash-only threshold score defined on the installation record or the account’s CIS Division, the account is flagged as cash-only during payment processing and on Control Central.
- Use **Credit Rating Months** to define the length of time the demerit remains in effect. This information is used to define the effective period of the credit rating demerit record.

FASTPATH:
For more information, refer to Account - Credit Rating.

The Send Letter collection event type causes a customer contact to be generated that, in turn, generates a letter. Enter the following **Parameters** for this type of event:

- Select the **Contact Class** used to categorize the customer contact.
- Use **Contact Type** to define the type of customer contact to create. The type of customer contact controls the type of letter that is generated.

> **NOTE:**
> **Letter creation is triggered via a customer contact.** You must set up a customer contact type for each type of letter you generate. You specify the necessary customer contact type on the collection event. Refer to Setting Up Letter Templates for more information.

The Cancel Budget collection event type cancel an account's budget plan (if the account is on such a plan). When a budget plan is cancelled, adjustments are issued to synchronize every service agreement's current balance with its payoff balance and each applicable SA's recurring charge amount (i.e., budget amount) is set to zero.

The Generic Algorithm collection event type causes the algorithm defined in the **Collection Event Alg** to be executed. You use this type of algorithm when the standard types of collection events won't do what you need done. If you haven't done so already, you must set up this algorithm in the system. To do this:

- Create a new algorithm (refer to Setting Up Algorithms).
- On this algorithm, reference an Algorithm Type that determines the algorithm that will be called when events of this type of activated. Click here to see the algorithm types available for this plug-in spot.

The Create To Do Entry collection event type causes a To Do entry to be issued. A good example of where this is used is when the collection event requires that the customer be called on the phone. Refer to The Big Picture of To Do Entries for more information about To Do entries (refer to the To Do type TD-CEVT for the type of To Do entry that's created).

The Start Severance Process type will start a severance process for every service agreement linked to the collection process. No parameters are needed for this type of event.

Enter a **Long Description** to fully describe the collection event type.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference CI_COLL_EVT_TYP.

Setting Up Collection Process Templates

Collection process templates define the collection events that will be executed when a collection criteria rule is violated. Open **Admin** > **Credit & Collection** > **Collection Process Template** > **Add** to define your collection process templates.

Description of Page

Enter a unique **Collection Process Template** and **Description** for the collection process template.

Select a **Cancel Criteria Algorithm** if your organization allows individual service agreements to be "removed" from a collection process regardless of the debt associated with all service agreements in the debt class. In other words, if your cancel criteria are based on the debt associated with ALL service agreements in a debt class DO NOT SPECIFY THIS ALGORITHM. If this algorithm is specified, it is executed by the collection process monitor when it detects that a credit has been applied to a service agreement linked to an active collection process. This algorithm will indicate if the specific service agreement that has been credited no longer has debt that warrants a collection process. Refer to How Are Collection Processes Cancelled for more information. If you haven't done so already, you must set up this algorithm in the system. To do this, create a new algorithm (refer to Setting Up Algorithms). On this algorithm, reference an Algorithm Type that "removes" a service agreement from a collection processes if the service agreement's debt so warrants. Click here to see the algorithm types available for this system event.

The **Response** grid contains an entry for every collection event that will be created when a collection process that references this template is created. The following information must be defined for each event:

Event Sequence controls the order in which the collection event types appear under the collection process template. The sequence number is system-assigned and cannot be changed. If you have to insert a collection event type between two existing templates, you'll have to remove the latter events, insert the new event, and then re-specify the removed events.

Collection Event Type Specify the type of collection event to be generated.

Days After Process Creation Specify the number of days after the creation of the collection process that the related collection event will be triggered. Refer to Calendar vs Work Days for a description of how this system uses this information to set the trigger date on the respective collection events.

- **FASTPATH:**
  For more information about collection event types, see Setting Up Collection Event Types. For more information about trigger dates, see Collection Event Trigger Date.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference CI_COLL_PROC_TM.

Setting Up Collection Classes

Every account has a collection class. This class is one of several fields that control the collection method applied to the account's debt. Open **Admin** > **Credit & Collection** > **Collection Class** to define your collection classes.

- **FASTPATH:**
  For more information about collection classes, see Designing Your Collection Classes.

Description of Page

Enter a unique **Collection Class** code and **Description** for each collection class.

Indicate which method is used to monitor the member accounts' unpaid debt:

- If you practice balance-forward accounting for accounts belonging to this collection class, select Collection, Severance & Write-Off. This method of collection is described throughout this chapter.
- If you practice open-item accounting for accounts belonging to this collection class, select Overdue. This method of collection is described under Defining Overdue Processing Options.
- If accounts belonging to this collection class are not subject to either of the above collection methods, select Not Eligible For Collection. Please be aware that these accounts will NOT be reviewed for overdue debt.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference CI_COLL_CL.

Setting Up Debt Classes

Every SA type has a debt class. This class is one of several fields that control the collection criteria applied to the service agreement's debt. Open **Admin** > **Credit & Collection** > **Debt Class** > **Add** to define your debt classes.

●

FASTPATH:
For more information about debt classes, see Designing Your Debt Classes.

Description of Page

Enter a unique **Debt Class** and **Description** for the debt class.

Turn on **Eligible for Collection** if service agreements belonging to this debt class have their debt monitored by the collection process. This should only be turned off if this debt cannot be collected, e.g., write-off debt.

The grid that follows 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** and **Algorithm** for each system event. You can set the **Sequence** to an arbitrary number as only one algorithm for each system event is allowed in this case.

WARNING:
These algorithms are typically significant system processes. The absence of an algorithm may prevent the system from operating correctly.

You can define algorithms for the following **System Events**:

System Event	Optional / Required	Description
Collection Process Cancellation Rule	Required if debt class is eligible for collection	This algorithm determines if a collection process can be canceled, and if so, it cancels it. Refer to How Are Collection Processes Cancelled for more information. Click here to see the algorithm types available for this system event.
Severance Process Cancellation Rule	Required if debt class is eligible for collection	This algorithm determines if a severance process can be canceled, and if so, it cancels it. Refer to How Are Severance Processes Cancelled for more information. Click here to see the algorithm types available for this system event.
Override Arrears Due To Pay Plan	Required if you use Pay Plans	This algorithm is called to temporarily override a customer's arrears using a pay plan's scheduled payments when the system looks at an account's debt from a credit & collections perspective (i.e., the ADM calls this algorithm before it subjects a customer's

debt to the collection criteria and when a pay plan is created). It does not actually change any data, but overlays the current arrears with the pay plan scheduled payments.

This algorithm is also called by the above algorithms when a pay plan is created in order to evaluate if the scheduled payments actually cover the arrears (if so, the collection / severance processes are cancelled). It is also called periodically by the ADM in order to establish if the current state of the pay plan still covers the arrears. Refer to How Pay Plans Affect The Account Debt Monitor.

Click here to see the algorithm types available for this system event.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference CI_DEBT_CL.

Setting Up Collection Class Controls

The topics in this section describe the windows on which you set up your collection class control information.

WARNING:
The information in this page is what controls how the system analyzes your customer's debt. The flexibility of this control information provides you with almost unlimited options. This is very powerful, but it requires careful analysis. Refer to Designing Your Collection Procedures**for more information**.

Collection Class Control - Main Information

The information on this transaction defines the conditions that will be checked by the Account Debt Monitor when it checks if an account has violated your debt criteria.

Open **Admin** > **Credit & Collection** > **Collection Class Control** > **Add** to define this information.

- **FASTPATH:**
  For more information about collection class control, refer to Designing Collection Class Controls.

Description of Page

Enter a unique **Collection Class Control** code and **Description** for the collection class control (CCC).

Enter the **CIS Division** to which the CCC's criteria applies.

Enter the **Collection Class** to which the CCC's criteria applies.

Enter the **Debt Class** to which the CCC's criteria applies.

Enter the **Currency Code** in which the CCC's criteria are denominated.

Use **Long Description** to further describe the CCC.

- **FASTPATH:**
  The information in the following grid is not intuitively obvious. Refer to Designing Collection Class Controls and Designing Your Collection Class Control Overrides for more information.

The grid which follows contains the conditions that are checked by the Account Debt Monitor (ADM) to determine the type of criteria (defined on the next tab) that will be applied against an account's debt. In other words - the ADM will check each condition (from highest to lowest **Priority**). The first condition that returns a value of true will cause the system to compare the account's debt against the debt criteria defined on the next tab.

Multiple conditions may be defined if different conditions result in a different type of debt thresholds (or a different type of collection process). The following fields are required for each condition:

Collection Condition Priority The priority controls the order in which the ADM checks if a collection condition applies (the lower the number, the higher the priority). Higher priorities are checked before lower priorities.

➤ **NOTE:**
The values for this field are customizable using the Lookup table. This field name is COLL_CAT_PRIO_ FLG.

Condition Algorithm Define the algorithm used to check if an account should be subject to the collection criteria defined on the next tab. If the algorithm returns a value of true (i.e., the condition is met), the ADM will compare the account's debt against the **Debt Criteria** (defined on the next tab) and start a collection process if the account has debt that violates these criteria.

You must have at least one collection condition; otherwise the system will not have criteria against which to compare the account's debt. This entry should have the lowest priority code and reference the "default" algorithm. If you haven't done so already, you must set up this algorithm in the system. To do this, create a new algorithm (refer to Setting Up Algorithms). On this algorithm, reference an Algorithm Type that references the "default" collection condition algorithm type (COLL COND DF).

If you have other conditions that should be checked before the default condition, you must create an entry for each in this grid. Each entry should have a priority consistent with your business requirements (and this priority should be higher than the default condition's priority). In addition, you should reference an algorithm that contains the conditions that will be checked to determine if the account should be subject to the debt criteria (defined on the next tab). The system is supplied with many additional algorithm types. In order to take advantage of them, you will need to create an algorithm (refer to Setting Up Algorithms). On this algorithm, reference an Algorithm Type that references one of the collection condition algorithm types. Click here to see the algorithm types available for this system event.

Where Used

Collection class controls contain the data that controls the Account Debt Monitor. Refer to How Does The Account Debt Monitor Work? for more information.

Collection Class Control - Debt Criteria

The information on this page defines the debt and age thresholds used by the Account Debt Monitor when it checks if an account has violated your acceptable levels of debt. Open **Admin** > **Credit & Collection** > **Collection Class Control** > **Search** and navigate to the **Debt Criteria** page to define this information.

➤ **NOTE:**

The information on this page is not intuitively obvious. Refer to Designing Collection Class Controls and Designing Your Collection Class Control Overrides for more information.

Description of Page

The **Debt Criteria** scroll contains an entry for each collection criteria algorithm defined on the **Main** tab. The following information appears

The **Collection Condition Priority** controls the order in which the Account Debt Monitor (ADM) checks if a collection condition applies. Higher priorities are checked before lower priorities.

The **Condition Algorithm** is called by the ADM to determine which collection criteria should be applied to the account's debt. If this algorithm returns a value of true (i.e., the condition is met), the ADM will compare the account's debt against the **Debt Criteria** defined below. If the account violates any criteria, a collection process will be started (using the respective **Collection Process Template**).

The grid that follows contains the debt age and amount of debt that must be violated by the account in order for the ADM to create a collection process template. The following fields should be defined:

Arrears Priority controls the order in which the arrears criteria will be checked by the Account Debt Monitor (the lower the number, the higher the priority). The first criteria, if any, that is met will cause a collection process to be created (using the **Collection Process Template**).

Collection Process Template If the Account Debt Monitor determines that the account's debt violates the corresponding criteria, it creates a collection process using the specified collection process template.

Arrears Amount When the Account Debt Monitor checks an account's debt, it determines if the account has debt older than "> Number of Days" (the next field) AND the debt exceeds "> Arrears Amount". If so, a collection process is started.

Days When the Account Debt Monitor checks an account's debt, it determines if the account has debt older than **Days** AND the debt exceeds **Arrears Amount**. If so, a collection process is started.

Where Used

Collection class controls contains the data that controls the Account Debt Monitor.

Designing Your Severance Procedures

The following matrix will help you design your severance procedures. When the matrix is complete, you're ready to set up the severance process control tables.

Notice that the matrix has two dimensions: one is dependent on severance criteria algorithms; the other is dependent on the SA type of the service agreement being severed. The number and type of SA types is dependent on how your organization sets up the SA type table (the SA types shown below are characteristic of those used by a simple utility).

SA Type	**Severance Criteria Algorithm:**	**Severance Criteria Algorithm:**
	Default	**Customer On Life Support**
Electric Residential		
Electric Commercial		
Gas Residential		
Gas Commercial		

Once you know the values of each dimension, you fill in each cell with its respective severance events. We've completed the sample matrix with some characteristic events.

SA Type	**Severance Criteria Algorithm:**	**Severance Criteria Algorithm:**
	Default	**Customer On Life Support**
Electric Residential	Create a 48-hour warning field activity. 2 days after completion, create a disconnect service field activity. Immediately after completion of the disconnect field activity, send a letter to the customer. 10 days after completion of disconnection, expire the service agreement.	Create a To Do entry asking a collection rep to call the customer. 5 days later, create a 72-hour warning field activity. 2 days after completion, create a To Do entry telling collection rep of impending life support cutoff. 3 days after completion of warning, create a disconnect service field activity AND generate a To Do entry informing a collection agent of such. Immediately after completion of the disconnect field activity, send a letter to the customer. 10 days after completion of disconnection expire the service agreement.
Electric Commercial	Create a 48-hour warning field activity. 2 days after completion, create a disconnect service field activity. Immediately after completion of the disconnect field activity, send a letter to the customer. 10 days after completion of disconnection, expire the service agreement.	N/A
Gas Residential	Create a 48-hour warning field activity. 2 days after completion, create a disconnect service field activity. Immediately after completion of the disconnect field activity, send a letter to the customer. 10 days after completion of disconnection, expire the service agreement.	Create a 48-hour warning field activity. 2 days after completion, create a disconnect service field activity. Immediately after completion of the disconnect field activity, send a letter to the customer. 10 days after completion of disconnection, expire the service agreement.
Gas Commercial	Create a 48-hour warning field activity. 2 days after completion, create a disconnect service field activity. Immediately after completion of the disconnect field activity, send a letter to the customer. 10 days after completion of disconnection, expire the service agreement.	N/A

Charitable Contribution	Expire service agreement	Expire service agreement

Once the matrix is complete, you determine the severance process templates needed to implement your severance procedures. The following table shows the severance process templates referenced in the previous section's matrix. Adjacent to each process are its events and an indication of when they are triggered.

Severance Process Template	Event Number	Severance Event Template	Dependent On Event(s)	Trigger Date Set To X Days After Completion Of Dependent Events
Utility severance - default	10	Field activity - 48 hour disconnect for non-payment warning	N/A - first event	0
	20	Field activity - disconnect for non-payment	10	2
	30	'Service has been disconnected' letter	20	0
	40	Expire service agreement	20	10
Electric life support residential severance	10	Generate delinquent life support customer To Do entry	N/A - first event	0
	20	Field activity - 72 hour disconnect for non-payment warning	10	5
	30	Generate impending life support cutoff To Do entry to C&C rep	20	2
	40	Field activity - cut for non-payment	20	3
	50	Service has been disconnected letter	40	0
	60	Expire service agreement	40	10
Just expire severance	10	Expire service agreement	N/A - first event	

If we extract each unique severance event type from the above table, we end up with the following:

Severance Event Template	Event Type
48-hour warning	Generate Field Activity - Disconnect Warning
72-hour warning	Generate Field Activity - Disconnect Warning
Disconnect for non payment	Generate Field Activity - Cut For Non-Payment
Delinquent life support customer	Create To Do Entry - C&C Rep Role
Impending life support cutoff	Create To Do Entry - C&C Rep Role
Service has been disconnected letter	Send Letter - Customer Contact Type is Disconnect Letter
Expire service agreement	Expire Service Agreement

WARNING:

The field activity types are NOT specified directly on the severance event type. Why? Because each service point linked to the service agreement being severed could necessitate a different type of field activity. Therefore, the system uses the type of service point, its state (e.g., connected, meter is off.) and the desired customer event (e.g., Disconnect Warning, Cut For Non-Payment) to determine which field activity type(s) to generate. Refer to Setting Up Field Activity Type Profiles for how to set up the specific disconnect field activity types for your various types of service points.

Now you're (almost) ready to set up your severance procedures.

Defining Severance Process Auto Cancellation Criteria

The topics in the section How Are Severance Processes Cancelled describe the two algorithms that play a part in the cancellation of a collection process. It also describes when to use what type of algorithm. Please read this section and then set up the appropriate cancellation criteria on your Debt Classes, and optionally, on your Severance Process Templates.

Designing Your Reconnection Procedures

If a customer pays for a service agreement after the service has been cut for non-payment AND BEFORE THE SA HAS BEEN EXPIRED, they need to be reconnected. Counter-intuitively, you must set up a severance process to initiate the field activities to reconnect service.

NOTE:
Why do you use a severance process to reconnect service? Because a severance process is nothing more than a series of events that take place one after another. Some of the events create field activities, others send letters, others create To Do entries. So, why not use a severance process? You just have to send different letters and perform different field activities.

WARNING:
The system will automatically create a reconnection process if a severance process is cancelled as a result of a payment (or other credits). Please note that this will only happen if you plug-in the appropriate post cancellation algorithm on your severance process templates. Refer to What Happens When A Severance Process Is Cancelled?**for more information**.

While you don't define the reconnect procedures for an SA type, we recommend you think about the reconnection steps for each of your SA types that can be disconnected for nonpayment. We've completed the sample matrix with some characteristic events.

SA Type	Steps
Electric Residential	Create a reconnect service field activity. Immediately after completion of the reconnect, send a letter to the customer.
Electric Commercial	Create a reconnect service field activity. Immediately after completion of the reconnect, send a letter to the customer.
Gas Residential	Create a reconnect service field activity.

	Immediately after completion of the reconnect, send a letter to the customer.
Gas Commercial	Create a reconnect service field activity. Immediately after completion of the reconnect, send a letter to the customer.

Once the matrix is complete, you determine the severance process templates needed to implement your reconnection procedures. Notice each SA type has the same reconnection steps. This means you just need one severance process. The following table shows this severance process template and its events.

Severance Process Template	Event Number	Severance Event Template	Dependent On Event(s)	Trigger Date Set To X Days After Completion Of Dependent Events
Reconnect	10	Field activity - reconnect service	N/A - first event	0
	20	Service has been reconnected letter	10	0

If we extract each unique severance event type from the above table, we end up with the following:

Severance Event Template	Event Type
Reconnect	Generate Field Activity - Reconnect
Service has been reconnected letter	Send Letter

WARNING:
The field activity types are NOT specified directly on the severance event type. Why? Because each service point linked to the service agreement being severed could necessitate a different type of field activity. Therefore, the system uses the type of service point, its state (e.g., connected, meter is off.) and the desired customer event (e.g., Disconnect Warning, Cut For Non-Payment) to determine which field activity type(s) to generate. Refer to Setting Up Field Activity Type Profiles for how to set up the specific disconnect field activity types for your various types of service points.

IMPORTANT:
If you want the system to automatically create a reconnection process if a customer pays after they have been cut, you must specify the appropriate post cancellation algorithm on your severance process templates.

And now you're ready to set up your severance (and reconnection) procedures.

Setting Up Severance Procedures

In the previous section, Designing Your Severance Procedures, we presented a case study that illustrated a mythical organization's severance procedures. In this section, we'll explain how to set up the control tables to implement these procedures:

Setting Up Severance Event Types

Severance event types control what is done by a given severance event. Open **Admin** > **Credit & Collection** > **Severance Event Type** > **Add** to define your severance event types.

- **FASTPATH:**
  For more information refer to Designing Your Severance Procedures.

Description of Page

Enter a unique **Severance Event Type** code and **Description** for the severance event type.

Enter the Severance **Event Type**. Permissible values are: Affect Credit Rating/Cash-Only, Send Letter, Generic Algorithm, Create To Do Entry, Create Field Activities, Expire Service Agreement. The following discussion describes the parameters that must be defined for each type of severance event.

The Affect Credit Rating/Cash-Only collection event type causes a credit rating demerit record to be linked to the account. This record is constructed using the following **Parameters**:

- account debt monitor.
- Use **Cash-Only Points** to define this event's affect on the account's cash-only score. This should be a positive number. When an account's cash-only score exceeds the cash-only threshold score defined on the installation record, the account is flagged as cash-only during payment processing and on Control Central.
- Use **Cash-Only Points** to define this event's affect on the account's cash-only score. This should be a positive number. When an account's cash-only score exceeds the cash-only threshold score defined on the installation record or the account’s CIS Division, the account is flagged as cash-only during payment processing and on Control Central.

The Send Letter severance event type causes a customer contact to be generated that, in turn, generates a letter. Enter the following **Parameters** for this type of event:

- Select the **Contact Class** used to categorize the customer contact.
- Use **Contact Type** to define the type of customer contact to create. The type of customer contact controls the type of letter that is generated.

The Generic Algorithm severance event type causes the algorithm defined in the **Sev. Event Algorithm** to be executed. You use this type of algorithm when the standard types of severance events won't do what you need done.

The Create To Do Entry severance event type causes a To Do entry to be created. Refer to The Big Picture of To Do Entries for more information about To Do entries (refer to the To Do type TD-SEVT for the type of To Do entry that's created).

The Generate Field Activities severance event type causes one or more field activities to be generated. Enter the following **Parameters** for this type of event:

- Select the **Customer Event** associated with the field activity. Valid values are: Cut For Non-Payment (CNP), Disconnect Warning (Disc Warn), Reconnect for Payment (Reconn), Start Service (Start) , Start/Stop Service (Start/Stop), Stop Service (Stop), Reread (Reread).
- The system uses the **Customer Event** to select the appropriate field activity type(s) from the field activity type profile associated with the service points linked to the service agreement.

WARNING:
The field activity types are NOT specified directly on the severance event type. Why? Because each service point linked to the service agreement being severed could necessitate a different type of field activity. Therefore, the system uses the

type of service point, its state (e.g., connected, meter off) and the desired customer event (e.g., Disconnect Warning, Cut For Non-Payment) to determine which field activity type(s) to generate. Refer to Setting Up Field Activity Type Profiles for how to set up the specific disconnect field activity types for your various types of service points.

The Expire Service Agreement severance event type causes the service agreement to be moved to the pending stop state (it also populates the service agreement's stop date). Refer to Finalizing Pending Stops for how the system will eventually stop the service agreement (and then final bill it). There are no parameters for this type of event.

NOTE:
Cut for non-payment field activities are special. The Expire Service Agreement severance event type also makes any "cut for non-payment" field activities created by earlier severance events available to stop service. Specifically, it changes the linkage type of the field activities from Severance Activity to Stop Activity. You can see a service agreement's service points' field activities and their respective linkage type on Service Agreement - Service Point.

Enter a **Long Description** to fully describe the severance event type.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference CI_SEV_EVT_TYPE.

Setting Up Severance Process Templates

Severance process templates define the severance events that will be executed when a service agreement is severed. Open **Admin** > **Credit & Collection** > **Severance Process Template** > **Add** to define your severance process templates.

Description of Page

Enter a unique **Severance Process Template** and **Description** for the severance process template.

If severance processes of this type should be automatically canceled when the customer pays the collection amount on the severance, turn on **Auto Cancel**. This switch would typically only be turned off for severance processes used to reconnect a cut service because you don't want such a reconnection process to be canceled when a payment is made. Refer to Designing Your Reconnection Procedures for more information.

In addition to turning on the **Auto Cancel** switch, specify a **Cancel Criteria Algorithm** if your organization allows a severance process to be cancelled regardless of the debt associated with all service agreements in the debt class. In other words, if your cancel criteria are based on the debt associated with ALL service agreements in a debt class DO NOT SPECIFY THIS ALGORITHM.

If the **Cancel Criteria Algorithm** is specified, it is executed when a credit is posted to the service agreement associated with a severance process. This algorithm will indicate if the service agreement no longer has debt that warrants a severance process. Refer to How Are Severance Processes Cancelled for more information. If you haven't done so already, you must set up this algorithm in the system. To do this, create a new algorithm (refer to Setting Up Algorithms). On this algorithm, reference an Algorithm Type that cancels a severance process if the service agreement's debt so warrants. Click here to see the algorithm types available for this plug-in spot.

If you wish to perform any special processes after a severance process is canceled, specify a **Post Cancel Algorithm**. This can be used to start a reconnection in case the severance process was canceled too late to stop the disconnection. If you haven't done so already, you must set up this algorithm in the system. To do this, create a new algorithm (refer to Setting Up Algorithms). On this algorithm, reference an Algorithm Type that cancels a severance process if the service agreement's debt so warrants. Click here to see the algorithm types available for this plug-in spot.

When a service agreement is to be severed due to non-payment, the system creates a severance process and links to it one or more severance events based on the **Event Types** entered here. The information in the scroll defines these events and the date on which they will be triggered. The following fields are required for each event:

Event Sequence controls the order in which the severance event types appear under the severance process template. The sequence number is system-assigned and cannot be changed. If you have to insert a severance event type between two existing templates, you'll have to remove the latter events, insert the new event, and then re-specify the removed events.

Severance Event Type Specify the type of severance event to be generated.

Dependent On Other Events Turn this indicator on if the trigger date of the event can only be determined after earlier events are complete. For example, you would turn this switch on for the event that initiates the field activity to disconnect service. Why? Because you only want to disconnect service after the preceding event that warned the client of impending disconnection is complete.

Days After Prev Response Specify the number of days after the completion / cancellation of the dependent events on which the severance event will be triggered. If this event is not dependent on the completion of other events, this field contains the number of days after the creation of the severance process that the related severance event will be triggered. Refer to Severance Event Dependencies and Trigger Date and Calendar vs Work Days for a description of how the system uses this information to set the trigger date on the respective severance events.

When the **Dependent On Other Events** switch is on, use the grid to define the events on which this event is dependent. If multiple events are specified in the grid, all such events must be completed or cancelled before the event will be triggered.

Sequence is system-assigned and cannot be specified or changed.

Dependent On Sequence Specify the sequence number of the severance event on which the above severance event depends.

FASTPATH:
For more information about severance event types, see Setting Up Severance Event Types. For more information about trigger dates, see Severance Event Dependencies & Trigger Date.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference CI_SEV_PROC_TMP.

Designing Your Write-Off Procedures

The design of your write-off procedures is relatively straightforward. Simply follow the instructions in the following topics.

Designing Your Write-Off Debt Classes

Multiple write-off debt classes are needed when you have different write-off procedures for different types of service agreements. If all service agreement debt is written-off the same way, then you'll have just one write-off debt class (call it Generic). However, if you're like many organizations, you will have multiple write-off debt classes. The following points will help you understand why:

- If you bill for 3 rd parties, you probably have different write-off debt classes for the 3 rd party service agreements. Why? Because you probably treat 3 rd party uncollectable debt differently from your own debt.
- You will need a separate write-off debt class for service agreements whose debt cannot be written off. Why? Because there is a switch on the write-off debt class control table that controls if service agreements in the write-off debt class are

eligible for write-off processing. Given that you will have some service agreements that hold debt that aren't eligible for write-off processing (e.g., service agreements that hold written-off debt and service agreements that overpayments), you will need at least one other write-off debt class.

- If you use the system to calculate charges for your organization's company usage, you'll need another write-off debt class (we refer to it as the "N/A" write-off debt class below). Why? Because all service agreements must have a write-off debt class, even those that will never have debt.

SA's Write-Off Debt Class	**Account's Collection Class**	**Account's Collection Class**
Normal W/O		
N/A		

Designing Write-Off Controls

Set up a matrix using the collection classes you designed when you were designing your collection procedures (Designing Your Collection Procedures).

SA's Write-Off Debt Class	**Account's Collection Class**	**Account's Collection Class**
	Residential	**Commercial/Industrial**
Normal Write Off		
N/A		

Each cell should have a "write-off control" that defines what to do when the system detects finaled debt that hasn't been paid. This is true even of the "N/A" write-off debt class. Why? Because you may want the system to write-down these stopped SAs when they have a small balance. For example, if you have a write-off service agreement that subsequently receives a partial payment that leaves a small amount owing, you probably want the system to generate a write-down adjustment (so that the write-off service agreement will close). We'll initially fill in the matrix for the "N/A' write-off debt class.

SA's Write-Off Debt Class	**Account's Collection Class**	**Account's Collection Class**
	Residential	**Commercial/Industrial**
Normal Write Off		
N/A	Attempt to reduce the SA's balance to zero using the following methods: Synchronize current balance with payoff balance. If the debt is < $10 and > $-1, write down the debt using a write-down adjustment.	Attempt to reduce the SA's balance to zero using the following methods: Synchronize current balance with payoff balance. If the debt is < $10 and > $-1, write down the debt using a write-down adjustment.

➤ **NOTE:**
If the Write Off Monitor encounters debt associated with a non-defined collection class and write-off debt class, it will issue an error.

For each cell that isn't designated as N/A, you need to answer the following questions:

- Are you allowed to transfer debt to other non-closed service agreements linked to the account? If so, you need to define the algorithm used to do such. Refer to Setting Up Write-off Control for more information about this algorithm.
- Are you allowed to write-down small amounts of debt (or small credits)? If so, you need to define the algorithm used to do such. Refer to Setting Up Write-off Control for more information about this algorithm.
- Should you refund credit balances with a check? If so, you need to define the algorithm to do such. Refer to Setting Up Write-off Control for more information about this algorithm.
- If debt remains after doing the above, how do you write it off (e.g., do you first refer the debt to a collection agency and only write it off after waiting 30 days)?

We'll fill in the above matrix with our assumptions:

SA's Write-Off Debt Class	**Account's Collection Class**	**Account's Collection Class**
	Residential	**Commercial/Industrial**
Normal Write-Off	Attempt to reduce the SA's balance to zero using the following methods: Attempt to transfer debt to another active service agreement linked to the account. If the debt is < $10 and > $-1, write down the debt using a write-down adjustment. If the debt is <= $-1, create an A/P adjustment to refund the credit to the customer. If debt remains: Highest priority: If customer has a non-cash deposit, create the non-cash deposit write-off process. Otherwise, create the default write-off process for residential debt.	Attempt to reduce the SA's balance to zero using the following methods: Attempt to transfer debt to another active service agreement linked to the account. If the debt / credit is < $10 and > $-10, write down the debt using a write-down adjustment. If the debt / credit is <= $-10, create an A/P adjustment to refund the credit to the customer. If debt remains: Highest priority: If customer has a non-cash deposit, create the non-cash deposit write-off process. Otherwise, create the default write-off process for commercial debt.
N/A	Attempt to reduce the SA's balance to zero using the following methods: Synchronize current balance with payoff balance. If the debt is < $10 and > $-1, write down the debt using a write-down adjustment. Because this debt class isn't eligible for further write-off processing, criteria used to process debt are not necessary.	Attempt to reduce the SA's balance to zero using the following methods: Synchronize current balance with payoff balance. If the debt is < $10 and > $-1, write down the debt using a write-down adjustment. Because this debt class isn't eligible for further write-off processing, criteria used to process debt are not necessary.

We can now use the information in the above matrix to design the necessary Write Off Process Templates and Write Off Event Types.

Designing Write Off Process Templates & Write Off Event Types

The following table shows the write-off process templates referenced in the previous section's matrix. Adjacent to each process are its events and an indication of when they are triggered.

Write Off Process Template	Write Off Event Type	Triggered X Days From Start Of Write Off Process

Residential	Refer to collection agency	0
	Letter notifying customer of referral	0
	Cancel collection agency referral	60
	Write off	60
Non-Cash Deposit Exists	To Do for non-cash deposit redemption	0
	To Do to highlight unpaid SA(s) still exist (and they will be reconsidered for write-off processing)	10
Commercial	Refer to collection agency	0
	Letter notifying customer of referral	0
	To Do to check up on collection agency's efforts	30
	Cancel collection agency referral	60
	Write off	60

If we extract each unique event type from the above table, we end up with the following:

Write Off Event Type	Event Type
Notification of write-off referral	Send letter - Debt referred to a collection agency
Refer to collection agency	Refer to collection agency
Cancel collection agency referral	Cancel collection agency referral
Write off	Write off
To Do for non-cash deposit redemption	Generate To Do - Redeem non-cash deposit
To Do to highlight unpaid SA(s) still exist	Generate To Do - SA(s) linked to a non-cash deposit remain unpaid (and will be reconsidered for write-off processing)
To Do to check up on collection agency's efforts	Generate To Do - Check up on collection agency's efforts

And now you're ready to set up your write-off procedures.

Setting Up Write-Off Procedures

In the previous section, Designing Your Write-Off Procedures, we presented a case study that illustrated a mythical organization's write off procedures. In this section, we'll explain how to set up the control tables to implement these procedures:

Setting Up Write Off Debt Classes

Every SA type has a write-off debt class. This class is one of several fields that control the write off criteria applied to the service agreement's debt. Select **Admin** > **Credit & Collection** > **Write Off Debt Class** to define your debt classes.

- **FASTPATH:**

For more information about debt classes, see Designing Your Write-Off Debt Classes.

Panel controls

To modify a write-off debt class, simply move to a field and change its value. To add a new write-off debt class, click + to insert a row, then fill in the information for each field. The following fields display:

Write Off Debt Class Code The unique identifier of the write off debt class.

Eligible for Write Off Indicates if service agreements belonging to this write off debt class are eligible for write-off processing. This should only be turned off if this debt cannot be written off, e.g., write off debt.

Description The description of the write off debt class.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference CI_WO_DEBT_CL.

Setting Up Write Off Event Types

Write-off event types control what is done by a given write-off event. Select **Admin** > **Credit & Collection** > **Write Off Event Type** > **Add** to define your write-off event types.

Description of Page

Enter a unique **Write Off Event Type Code** and **Description** for the write-off event type.

Enter the **Write Off Event Type**. Permissible values are: Affect Credit Rating/Cash-Only, Cancel Agency Referral, Generic Algorithm, Refer to Agency, Send Letter, Create To Do Entry, Write Off using Distrib Code, and Write Off using SA Type. The following discussion describes the parameters that must be defined for each type of write-off event.

The Affect Credit Rating/Cash-Only write-off event type causes a credit rating demerit record to be linked to the account. This record is constructed using the following **Parameters**:

- Use **Affect Credit Rating By** to define this event's affect on the account's credit rating. This should be a negative number. An account's credit rating is equal to the beginning credit rating amount defined on the installation record or the account’s CIS Division plus the sum of credit rating demerits that are currently in effect. When an account's credit rating is less than the credit rating threshold defined on the installation record or the account’s CIS Division, the account's credit rating is displayed as an alert on Control Central.
- Use **Affect Cash-Only Score By** to define this event's affect on the account's cash-only score. This should be a positive number. When an account's cash-only score exceeds the cash-only threshold score defined on the installation record or the account’s CIS Division, the account is flagged as cash-only during payment processing and on Control Central.
- Use **Months Affecting Credit Rating** to define the length of time the demerit remains in effect. This information is used to define the effective period of the credit rating demerit record.

FASTPATH:
For more information, refer to Account - Credit Rating.

The Cancel Agency Referral event type will cancel previous collection agency referrals. No parameters are needed for this type of event.

The Generic Algorithm write-off event type causes the algorithm defined in the **Generic Algorithm** to be executed. You use this type of algorithm when the standard types of write off events won't do what you need done. If you haven't done so already, you must set up this algorithm in the system. To do this:

- Create a new algorithm (refer to Setting Up Algorithms).
- On this algorithm, reference an Algorithm Type that determines the algorithm that will be called when events of this type are activated. Click here to see the algorithm types available for this plug-in spot.

The Refer to Agency event type will refer the debt associated with the process' SAs to a collection agency. You must supply the **Agency Selection Algorithm** that is used to determine the collection agency associated with the referral. If you haven't done so already, you must set up this algorithm in the system. To do this:

- Create a new algorithm (refer to Setting Up Algorithms).
- On this algorithm, reference an Algorithm Type that determines the collection agency to which bad debt should be referred. Click here to see the algorithm types available for this plug-in spot.

> **NOTE:**
> **Letters.** You must set up a customer contact type for each type of letter you generate. You specify the necessary customer contact type on the write off event type. Refer to Setting Up Letter Templates for more information.

The Send Letter write-off event type causes a customer contact to be generated that, in turn, generates a letter. Enter the following parameters for this type of event:

- Select the **Contact Class** used to categorize the customer contact.
- Use **Contact Type** to define the type of customer contact to create. The type of customer contact controls the type of letter that is generated.

The Create To Do Entry write-off event type causes a To Do entry to be issued. Refer to The Big Picture of To Do Entries for more information about To Do entries (refer to the To Do type TD-WOEVT for the type of To Do entry that's created).

The Write Off using Distrib Code event type causes bad debt to be written off in accordance with the distribution codes associated with the financial transactions that caused the debt in the first place. Use this method if, for example, you want to write-off revenue differently than you write-off liabilities. When this type of event is activated, the system accumulates the distribution codes from GL details associated with recent financial transactions linked to each write-off service agreement. When the system has accumulated enough distribution codes (i.e., where the amount associated with the distribution code equals or exceeds the amount to write off), the debt will be transferred to a new or existing write-off service agreements. The type of service agreements to which the debt is transferred is defined on the distribution codes. Refer to Setting Up Distribution Codes for more information.

The Write Off using SA Type event type causes all debt associated with the process' SAs to be transferred to a write-off service agreement linked to the account. Enter the following **Parameters** for this type of event:

- **CIS Division / SA Type** is the type of write-off service agreement to which the debt will be transferred. Note well,
  - The system will reuse an existing service agreement if an active SA of this type is already linked to the account; otherwise the system will create a new service agreement of this type.
  - The adjustment type used to set the offending service agreement's current balance equal to its payoff balance is defined on the write-offable SA Type. Refer to SA Type - Main Information for more information.
  - The adjustment type used to transfer the delinquent debt to the write-off service agreement is defined on the write off SA type. Refer to SA Type - Detail for more information.

Enter a **Comment** to fully describe the write-off event type.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference CI_WO_EVT_TYP.

Setting Up Write Off Process Templates

Write-off process templates define the write-off events that will be executed when a write-off criteria rule is violated. Select **Admin** > **Credit & Collection** > **Write Off Process Template** > **Add** to define your write-off process templates.

Description of Page

Enter a unique **Write Off Process Template** code and **Description** for the write-off process template.

The rows in the following grid define the events that will be created when a write off process is created using this template. The following fields display:

Event Sequence scontrols the order in which the write-off event is executed. The sequence number is system assigned and cannot be changed. If you need to insert a write-off event between two existing events, you must remove the latter events, insert the new event, and then re-enter the removed events.

Write-off Event Type Code Specify the type of write-off event to be generated. The event type's description is displayed adjacent.

Days After Process Creation Specify the number of days after the creation of the write-off process that the related write-off event will be triggered. Refer to Calendar vs Work Days for a description of how this system uses this information to set the trigger date on the respective write-off events.

- **FASTPATH:**
  For more information about write-off event types, see Setting Up Write Off Event Types. For more information about trigger dates, see Write-off Event Trigger Date.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference CI_WO_PROC_TMPL.

Setting Up Write Off Control

Write-off controls define how the system handles finaled, unpaid debt belonging to a given collection class and write off debt class.

Write Off Control - Main

Select **Admin** > **Credit & Collection** > **Write Off Control** > **Add** to define basic information about a write-off control. After entering basic information, navigate to the **Criteria** tab to define the type of write-off process to start when given criteria are met.

- **FASTPATH:**
  For more information about write-off control, refer to Designing Write-Off Controls.

Panel controls

Enter a **Write Off Control** code and **Description** for the write-off control (WOC).

Enter the **Collection Class** to which the WOC applies.

Enter the **Write Off Debt Class Code** to which the WOC applies.

Enter general **Comments** to further describe the WOC.

Define the **Synch All Algorithm** used by the system to generate adjustments that cause current balance to equal payoff balance on the service agreements to be written off. This type of algorithm is issued before you actually start a write-off process as current balance is meaningless at write-off time (the customer owes you the payoff balance). If you haven't done so already, you must set up this algorithm in the system. To do this:

- Create a new algorithm (refer to Setting Up Algorithms).
- On this algorithm, reference an Algorithm Type that synchronizes current and payoff balances. Click here to see the algorithm types available for this plug-in spot.

Define the **Debt Transfer Algorithm** used by the system when it attempts to transfer the unpaid debt to another active service agreement linked to the stopped SA's account. If you haven't done so already, you must set up this algorithm in the system. To do this:

- Create a new algorithm (refer to Setting Up Algorithms).
- On this algorithm, reference an Algorithm Type that transfers unpaid balances. Click here to see the algorithm types available for this plug-in spot.

Define the **Write Down Algorithm** used by the system when it attempts to write-down small debt and/or credit balances. If you haven't done so already, you must set up this algorithm in the system. To do this:

- Create a new algorithm (refer to Setting Up Algorithms).
- On this algorithm, reference an Algorithm Type that writes down small amounts. Click here to see the algorithm types available for this plug-in spot.

Define the **Credit Refund Algorithm** used by the system when it refunds a credit balance to a customer. If you haven't done so already, you must set up this algorithm in the system. To do this:

- Create a new algorithm (refer to Setting Up Algorithms).
- On this algorithm, reference an Algorithm Type that refunds credit balances. Click here to see the algorithm types available for this plug-in spot.

Write Off Control - Criteria

Select **Admin** > **Credit & Collection** > **Write Off Control** > **Search** and navigate to the **Criteria** page to define the type of write-off process to start when given criteria are met.

> **FASTPATH:**
> The following information is not intuitively obvious. Refer to Designing Write-Off Controls for more information.

Panel controls

The information in the grid defines the write-off process to be executed for debt belonging to the previously defined collection class and write off debt class. The type of write-off process may differ depending on some condition. For example, you may have a different write-off process if the customer has a non-cash deposit. You must have at least one entry in this collection otherwise the system will not start a write-off process. This entry should have the lowest priority code and should reference a **Write Off Criteria Algorithm** that references the WO CRIT DFLT the algorithm type.

The following fields display:

Priority controls the order in which the criteria will be checked by the Write Off Monitor (higher priorities are checked before lower priorities). The first criteria algorithm that is met (i.e., returns a value of *True*) will cause the associated write-off process to be initiated.

NOTE:
The values for this field are customizable using the Lookup table. This field name is CRIT_PRIO_FLG. Be aware that this field is used for multiple tables: Collection Class Control, Severance Criteria, and Write Off Control .

Write Off Criteria Algorithm The Write Off Monitor checks if the condition defined by the W/O Condition Algorithm applies to the account whose debt is being analyzed. If a condition is met, a write-off process is created using the associated write-off process template.

If you haven't done so already, you must set up this algorithm in the system. To do this:

- Create a new algorithm (refer to Setting Up Algorithms).
- On this algorithm, reference an Algorithm Type that determines if a customer's bad debt should be processed using the associated **Write Off Process Template**. Click here to see the algorithm types available for this plug-in spot.

IMPORTANT:
You must have at least one entry in this grid otherwise the system will not start a write-off process. This entry should have the lowest priority code and should reference a **W/O Criteria Algorithm** that references the WO CRIT DFLT algorithm type.

Write Off Process Template If the Write Off Monitor determines the condition defined by the w/o condition algorithm applies, a write-off process is created using the associated write-off process template.

Where Used

Write-off controls contain the data that controls the Write Off Monitor. Refer to How Does The Write-Off Monitor Work? for more information.

Setting Up Collection Agencies

You must set up a collection agency for each such organization to which you refer delinquent debt. To define a collection agency, select **Admin** > **Credit & Collection** > **Collection Agency.**

Description of Page

Enter an easily recognizable **Collection Agency** code and **Description** for each collection agency.

A collection agency must be associated with a Person. Choose the **Person ID** of the organization from the prompt.

FASTPATH:
Information about how to set up persons is discussed in Maintaining Persons.

Turn on the **Active** switch if the collection agency is actively receiving referrals.

Specify the **Batch Control** that's used to route new and cancelled referrals to the collection agency. The batch control's description is displayed adjacent.

Where Used

Collection agencies get assigned to collection agency referrals when the collection agency referral background process executes. Refer to How Do Collection Agency Referrals Work? for more information.

Setting Up Feature Configuration

You must set up a Feature Configuration if you use the champion / challenger functionality.

The following describes the various **Option Types** that must be defined:

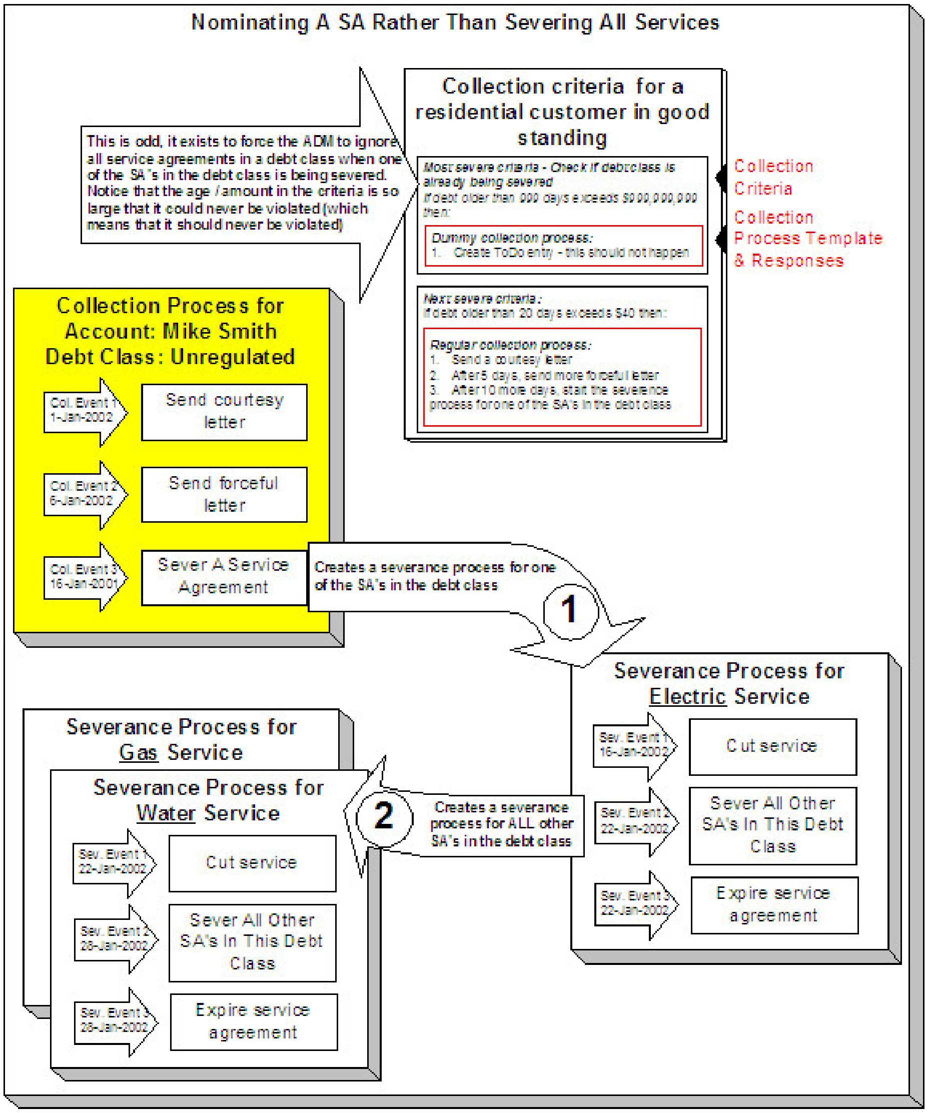
Champion Template$Challenger Template$Percentage(1-100). You need only set up options of this type if your implementation implements Champion / Challenger functionality. Options of this type are entered in the format A$B$nnn where A is the collection process template of the champion template, B is the collection process template of the challenger template, and C is the percent of the time that the system should create the challenger template. The collection monitor uses this option to override the champion collection process template X% of the time with the challenger template. You may enter any number of these options (but only one per Champion Template).

How To

The contents of this section describe how to set up various credit and collections scenarios.

How To Nominate A Single Service To Sever (Rather Than Sever Everything That's In Arrears)

Some organizations that offer multiple services do not sever all services when the customer falls into arrears. Rather, they nominate a single service agreement to sever in the hopes that the lack of service will cause the customer to remit payment. The following diagram illustrates the control tables values required to implement this type of requirement.


Nominating A SA Rather Than Severing All Services
This is odd, it exists to force the ADM to ignore all service agreements in a debt class when one of the SA's in the debt class is being severed. Notice that the age / amount in the criteria is so large that it could never be violated (which means that it should never be violated)
Collection criteria for a residential customer in good standing
Most severe criteria - Check if debtclass is already being severed
If debt older than 999 days exceeds $999,999,999 then:
Dummy collection process:
1. Create ToDo entry - this should not happen
Next severe criteria:
If debt older than 20 days exceeds $40 then:
Regular collection process:
1. Send a courtesy letter
2. After 5 days, send more forceful letter
3. After 10 more days, start the severence process for one of the SA's in the debt class
Collection Criteria
Collection Process Template & Responses
Collection Process for Account: Mike Smith Debt Class: Unregulated
Col. Event 1 1-Jan-2002
Send courtesy letter
Col. Event 2 6-Jan-2002
Send forceful letter
Col. Event 3 16-Jan-2001
Sever A Service Agreement
Creates a severance process for one of the SA's in the debt class
1
Severance Process for Electric Service
Sev. Event 1 16-Jan-2002
Cut service
Sev. Event 2 22-Jan-2002
Sever All Other SA's In This Debt Class
Sev. Event 3 22-Jan-2002
Expire service agreement
2
Creates a severance process for ALL other SA's in the debt class
Severance Process for Gas Service
Severance Process for Water Service
Sev. Event 1 22-Jan-2002
Cut service
Sev. Event 2 28-Jan-2002
Sever All Other SA's In This Debt Class
Sev. Event 3 28-Jan-2002
Expire service agreement

The following important concepts are illustrated above:

- The collection process's last event does NOT sever all services. Rather, it calls an algorithm that selects a single service to sever. A base package algorithm allows you to define the primary service to sever and a secondary service to sever (if the customer does not have the primary service). If you sold electricity and gas, you would probably define the primary service as electricity and the secondary as gas (because electricity is easier to cutoff than gas).
- The severance process that is started for the primary service cuts service. If the customer doesn't remit what is owed, the second severance event calls an algorithm that severs all other service agreements in the debt class.
- Because you are nominating a single service to sever, you must set up a special value in collection class control to force the ADM to ignore all service agreements in a debt class when one of the SAs in the debt class is being severed. Notice that the age / amount in the criteria is so large that it could never be violated.
- In addition, because the entire debt class must no longer be in arrears to stop the collection and severance processes, you must plug-in the appropriate collection process and severance process cancellation criteria on the debt class. Refer to How Are Collection Processes Cancelled and How Are Severance Processes Cancelled for more information about how these algorithms are used. Also note, you do not need service agreement cancellation criteria defined on your collection process templates and severance process templates (because cancellation is controlled at the debt class level).

Defining Device Options

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 from the **Admin > Device > Command Set**.

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.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference D1_COMMAND_SET.

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 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 Process 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 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.

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
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

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...
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 from the **Admin** > **Device** > **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.
- **Service Provider:** 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.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference D1_SPR.

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 from the **Admin** > **Device** > **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.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference D1_MANUFACTURER.

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:

- **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.).
- **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
Interval Channel Type - Physical	Provides the configuration for a physical interval channel (e.g. interval size). This is recommended for measuring components that measure consumptive usage.	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 (e.g. 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.	D1-IntrvlChanTypPhysSubtractiv
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
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). Note: these measuring component types can also be used to model interval data that is received with a large interval size (e.g. 24 hours).	D1-AutoReadRegisterType

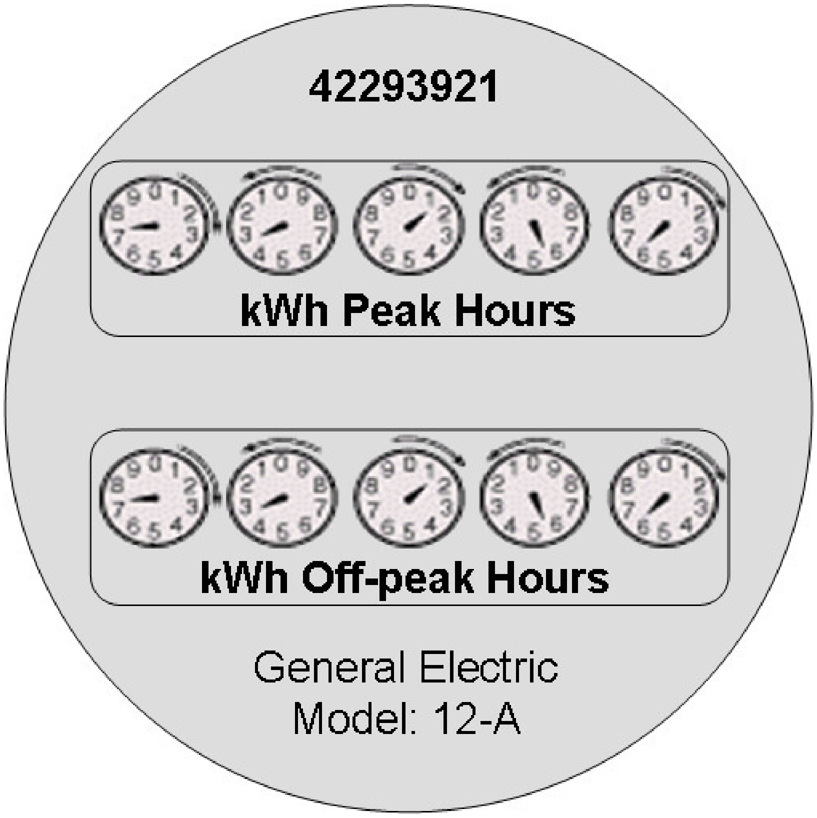
Refer to Configuring an Out-of-the-box Aggregation for more information about aggregation configuration

Measuring Component Type Value Identifiers

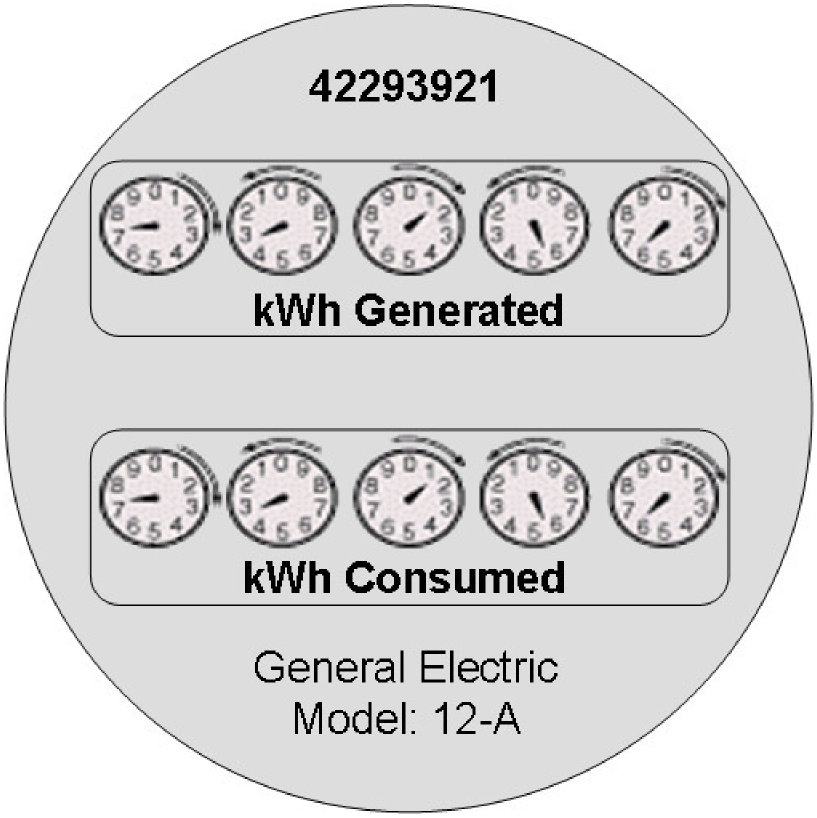
Measuring components are configured to measure specific types of quantities this is defined by the list of value identifiers on the measuring component type. Up to eleven value identifiers can be specified. The primary measured quantity should be identified using the Value Identifier Type of Measurement. An additional ten derived values can be computed based on the primary 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 Understanding Devices, Measuring Components, and Device Configurations 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).



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.



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 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 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 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.

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 from the **Admin** > **Device** > **Measuring Component** > **Search.**

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.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference D1_MEASR_COMP_TYPE.

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.

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 from the **Admin** > **Device** > **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.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference D1_DVC_TYPE.

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)

When creating a new device type there are the following options:

Name	Details	Business Object
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."	D1-ItemType
Communications Component Device Type	These devices provide remote collection of consumption data measured by a manual meter to which they are attached. These devices are associated to a head end system.	D1-CommunicationCompMeterType

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 are a list of business object options that are defined on the device business object and their impact on system processing:

- **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.

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 from the **Admin > Device > 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.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference D1_DVC_TYPE.

Defining Device Installation Options

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. A service point type may also have one or more service agreement types (SA Types) that define the types of service agreements that may be linked to service points of this type.

The "Service Point Category" field defines 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.

> **NOTE:** In Oracle Utilities Customer To Meter, Service Point Types are maintained via Merged Maintenance.

Configuring Service Point Types

This portal is used to display and maintain a Service Point Type.

Refer to Understanding Service Point Types for more information.

You can access the portal from **Admin** > **Geographic** > **Service Point Type**.

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.

> **NOTE:** In Oracle Utilities Customer To Meter, Service Point Types are maintained via Merged Maintenance.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference D1_SP_TYPE.

Measurement Cycles

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.

For a deeper functional understanding, refer to the About Route Management section of the *Oracle Utilities Meter Data Management / Smart Grid Gateway Business User Guide*.

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 from the **Admin > Device Installation > 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 Type:** 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.
- **Measurement Cycle Schedule List:** This zone lists the measurement cycle schedules related to the current measurement cycle.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference D1_MSRMT_CYC.

Measurement Cycle and Bill Determinants

The system can be configured to periodically push bill determinants to subscribing systems. 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).

When the Pending SP/Measurement Cycle Schedule Route records are processed by the D1-PSPSR batch, rather than create a handheld download activity, the algorithm can create a usage transaction (usage transactions are transactions that cause bill determinants to be calculated for the Service Point's Usage Subscription).

If the implementation needs to both manually read the meter and push bill determinants, both algorithms would be plugged in on the SP/Measurement Cycle Schedule Route business object.

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, etc. 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. Custom algorithms can be configured to do any additional necessary work, such as creating a "Meter Read Download" activity. This custom algorithm would be configured as an Enter algorithm on the "Complete" state of the SP/ Measurement Cycle Schedule Route business object.

For a deeper functional understanding, refer to the About Route Management section of the *Oracle Utilities Meter Data Management / Smart Grid Gateway Business User Guide*.

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 Data Management / Smart Grid Gateway 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 from the **Admin > Device Installation > 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 Type Schedule:** This zone provides information about the selected Measurement Cycle Schedule.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference D1_MSRMT_CYC_SCHED.

Setting Up Metered Premise Options

This section describes tables that must be set up before you can define premises.

Setting Up Meter Read Instructions

When you set up a premise you may define instructions to be supplied to the individuals who read the meters located at the premise. To define meter read instruction codes, open **Admin** > **Geographic** > **Meter Read Instruction.**

Description of Page

Enter a unique **Meter Read InstructionCode** and **Description** for every meter read instruction.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference CI_MR_INSTR.

Setting Up Meter Read Warnings

When you set up a premise you may define warnings to be supplied to the individuals who read the meters located at the premise. To define meter read warning codes, open **Admin** > **Geographic** > **Meter Read Warning.**

Description of Page

Enter a unique **Meter Read Warning** and a **Description** for every meter read warning.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference CI_MR_WARN.

Setting Up Consumption Estimation Parameters

The system estimates consumption under the following situations:

- At billing time, when "real" consumption cannot be computed due to the lack of a meter read, the system estimates how much the customer used assuming the following conditions are true:
  - The service agreement allows estimation.
  - The rate schedule allows estimation.
  - The bill cycle schedule (if one is being used) allows estimation.
  - The register is not a peak register (this is an artificial constraint, but most organizations do not estimate peak consumption). Note that the base product estimation algorithms may attempt to estimate peak registers.
  - The register is subtractive. This is because only subtractive registers are self-correcting. For example, assume we estimate a reading of 100 and the next real read is 102, the customer will only be charged for 2 units. But consider what would happen if we estimated a consumptive register, the consumption associated with the next real read will be billed in its entirety.

- When a meter read is added, the system verifies that the resultant consumption is congruous with historical consumption trends (i.e., high / low checks are performed).

This section describes the tables that must be set up to estimate consumption.

Defining Device Measurement Options

Units of Measure

Understanding Units of Measure

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.

NOTE: In Oracle Utilities Customer To Meter, Units of Measure are maintained via Merged Maintenance.

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 from the **Admin** > **Rates** > **Units 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.

NOTE: In Oracle Utilities Customer To Meter, Units of Measure are maintained via Merged Maintenance.

Where Used

Follow this link to open the Application Viewer data dictionary where you can view the tables that reference D1_UOM and UOM.

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).

> **NOTE:** In Oracle Utilities Customer To Meter, Service Quantity Identifiers are maintained via Merged Maintenance.

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 from **Admin** > **Rates** > **Service Quantity Identifier**.

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

- **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.

> **NOTE:** In Oracle Utilities Customer To Meter, Service Quantity Identifiers are maintained via Merged Maintenance.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference D1_SQI and SQI.

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.

➤ **NOTE:** In Oracle Utilities Customer To Meter, Time of Use periods are maintained via Merged Maintenance.

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 from **Admin** > **Rates** > **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.

➤ **NOTE:** In Oracle Utilities Customer To Meter, Time of Use periods are maintained via Merged Maintenance.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference D1_TOU and TOU.

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.

➤ **NOTE:** In Oracle Utilities Customer To Meter, Service Types are maintained via Merged Maintenance.

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 from **Admin** > **General** > **Service Type**.

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.

NOTE: In Oracle Utilities Customer To Meter, Service Types are maintained via Merged Maintenance.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference D1_SVC_TYPE and SERVICE TYPE.

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 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 from the **Admin** > **General** > **Factor** > **Search.**

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.
- **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 hyperlink to see the Factor Value in more detail on the Factor Value portal.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference D1_FACTOR.

Initial Measurement Data

Configuring the Initial Measurement Algorithms

The behavior of initial measurement data processing can be adjusted by updating parameters in several key algorithms in the lifecycle.

> **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):
  - 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 the IMD Seeder 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):
  - 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.

> **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** > **Integration** > **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):

- 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). Specifically the "Create Measurement Log on Update (Y/N)" parameter should be set to "Y". As mentioned creating these logs has a performance impact and it is not recommended for use on initial load.
- **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.

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.

Defining VEE Options

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 Data Management/Smart Grid Gateway 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 from **Admin** > **VEE** > **Exception Type** > **Search.**

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

Where Used

Follow this link to open the data dictionary where you can view the tables that reference D1_EXCP_TYPE.

VEE Groups

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 Data Management / Smart Grid Gateway Business User Guide*.

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 from the **Admin** > **VEE** > **VEE Group** > **Search** 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
- **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

Where Used

Follow this link to open the data dictionary where you can view the tables that reference D1_VEE_GRP.

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 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 Data Management / Smart Grid Gateway 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 Data Management / Smart Grid Gateway Business User Guide*.

You can access the portal from the **Admin** > **VEE** > **VEE Rule** > **Search**. 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.
- **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.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference D1_VEE_RULE.

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.
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.
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.
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. ***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.
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.
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

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.

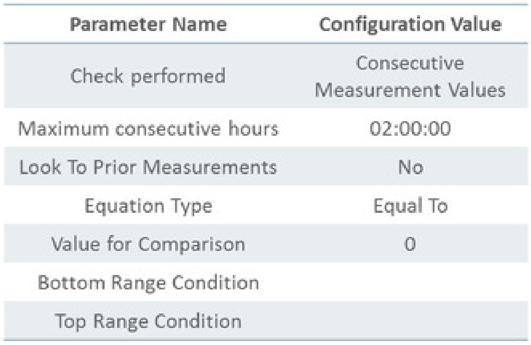
Business Object

D2-ConsecutiveIntervalCheck

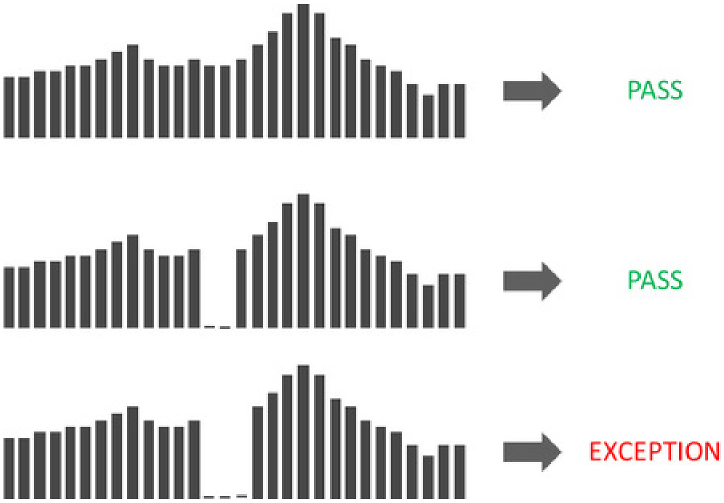
Example Scenarios

Below are some example scenarios that can be achieved based on configuration of this rule.

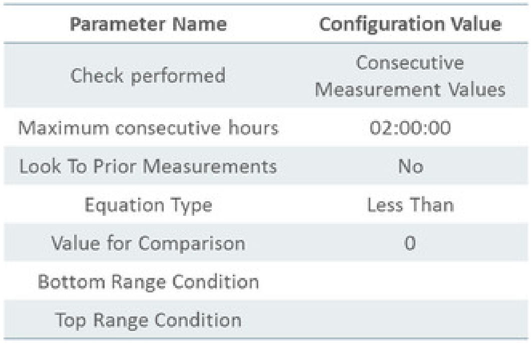
Scenario 1: Rule configured to fail for zero values



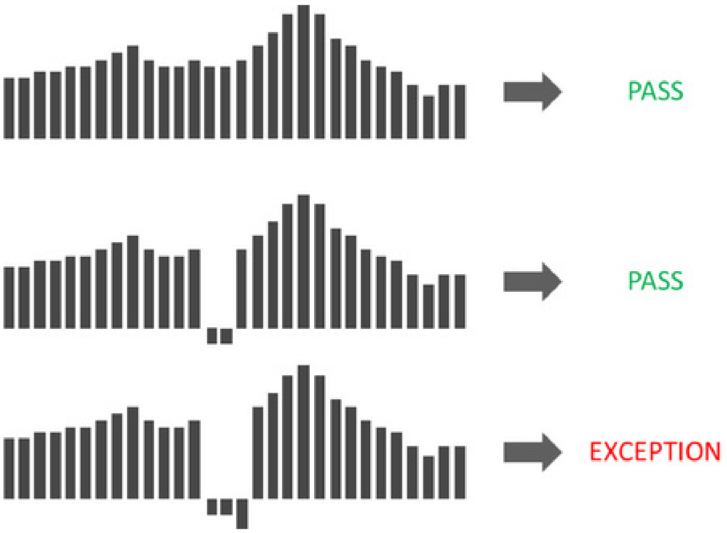
Parameter Name	Configuration Value
Check performed	Consecutive Measurement Values
Maximum consecutive hours	02:00:00
Look To Prior Measurements	No
Equation Type	Equal To
Value for Comparison	0
Bottom Range Condition	
Top Range Condition	



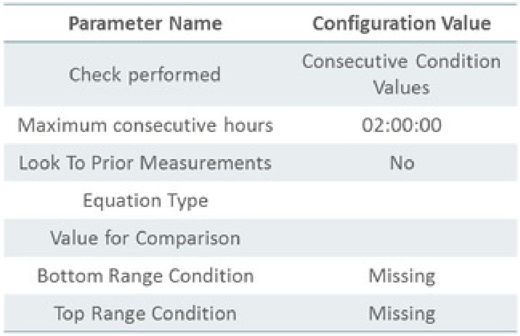
Scenario 2: Rule configured to fail for negative values



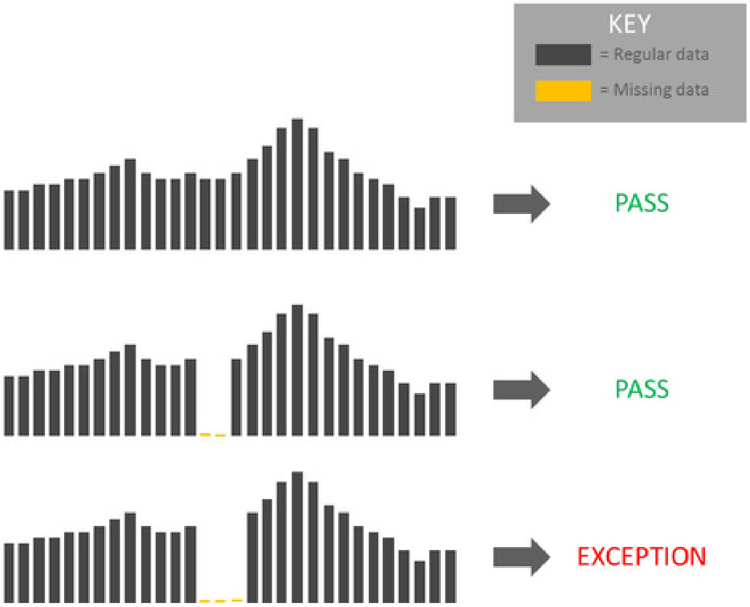
Parameter Name	Configuration Value
Check performed	Consecutive Measurement Values
Maximum consecutive hours	02:00:00
Look To Prior Measurements	No
Equation Type	Less Than
Value for Comparison	0
Bottom Range Condition	
Top Range Condition	



Scenario 3: Rule configured to fail for missing condition codes

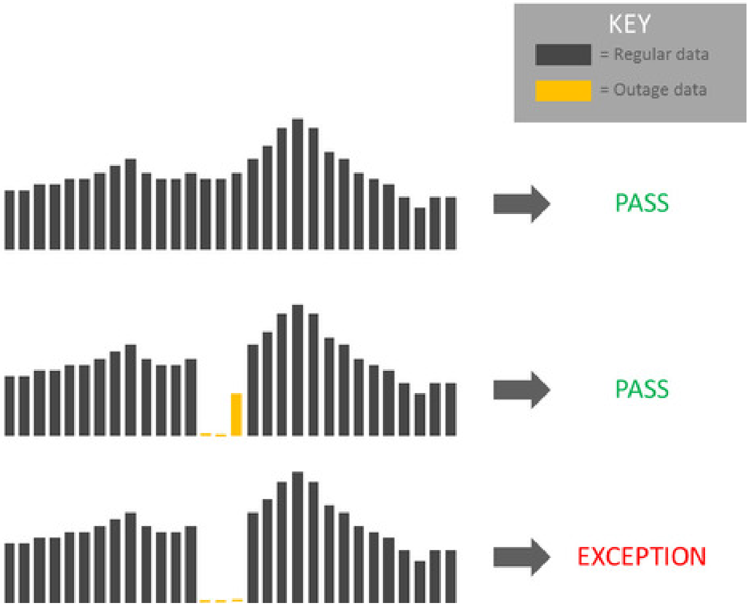


Parameter Name	Configuration Value
Check performed	Consecutive Condition Values
Maximum consecutive hours	02:00:00
Look To Prior Measurements	No
Equation Type	
Value for Comparison	
Bottom Range Condition	Missing
Top Range Condition	Missing



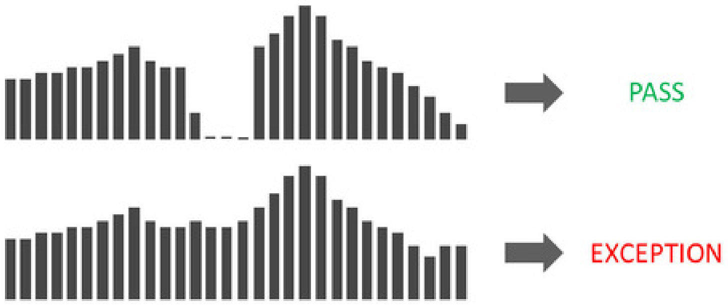
Scenario 4: Rule configured to fail for outage condition codes only if its zero

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



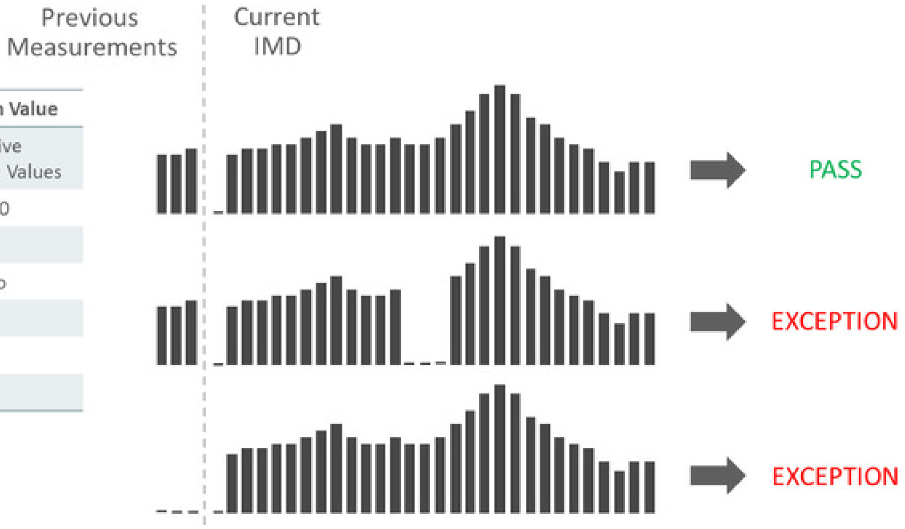
Scenario 5: Rule configured to fail if a long period of consecutive water values are greater than 0.1

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

Overview

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.

Business Object

D1-DuplicateIMDCheck

Dynamic Comparison Validation

Overview

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

Please note, this rule does not query large sets of historical data and will perform very well.

➤ **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.

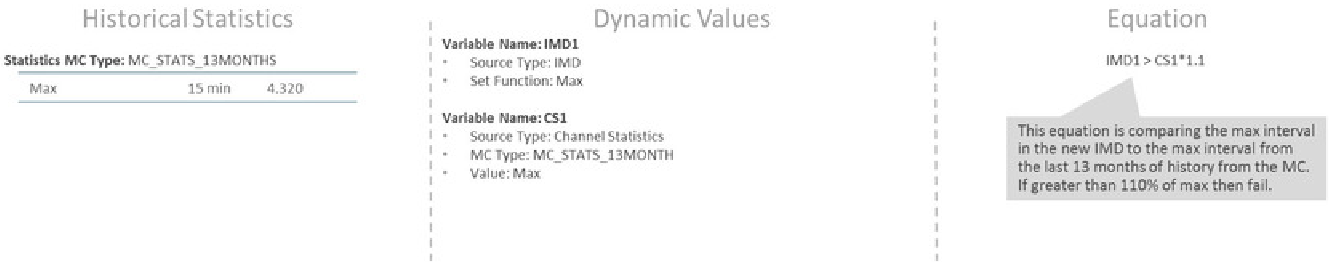
Business Object

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"



Historical Statistics

Statistics MC Type: MC_STATS_13MONTHS

Max	15 min	4.320

Dynamic Values

Variable Name: IMD1
- Source Type: IMD
- Set Function: Max

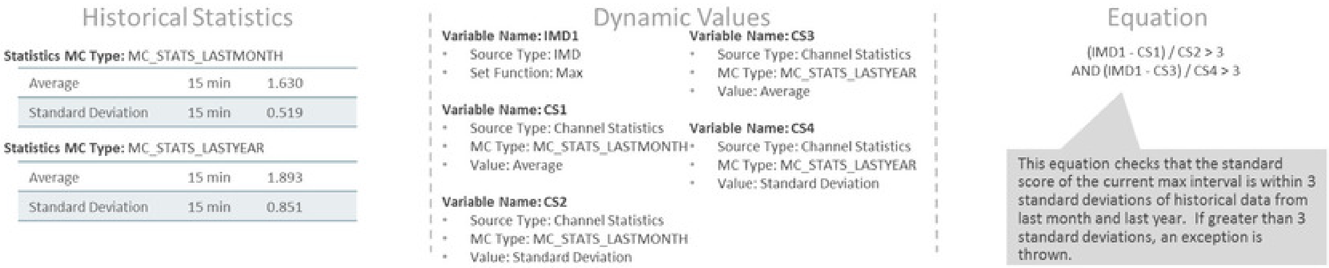
Variable Name: CS1
- Source Type: Channel Statistics
- MC Type: MC_STATS_13MONTH
- Value: Max

Equation

IMD1 > CS1*1.1

This equation is comparing the max interval in the new IMD to the max interval from the last 13 months of history from the MC. If greater than 110% of max then fail.

Example 2: High Usage



Historical Statistics

Statistics MC Type: MC_STATS_LASTMONTH

Average	15 min	1.630
Standard Deviation	15 min	0.519

Statistics MC Type: MC_STATS_LASTYEAR

Average	15 min	1.893
Standard Deviation	15 min	0.851

Dynamic Values

Variable Name: IMD1
- Source Type: IMD
- Set Function: Max

Variable Name: CS1
- Source Type: Channel Statistics
- MC Type: MC_STATS_LASTMONTH
- Value: Average

Variable Name: CS2
- Source Type: Channel Statistics
- MC Type: MC_STATS_LASTMONTH
- Value: Standard Deviation

Variable Name: CS3
- Source Type: Channel Statistics
- MC Type: MC_STATS_LASTYEAR
- Value: Average

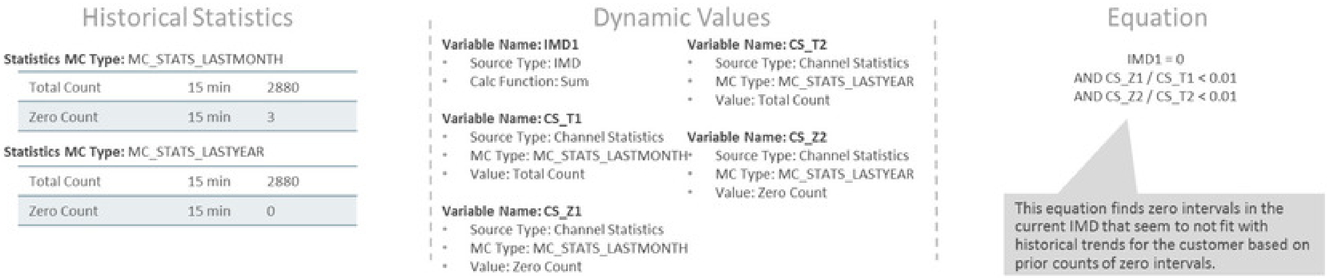
Variable Name: CS4
- Source Type: Channel Statistics
- MC Type: MC_STATS_LASTYEAR
- Value: Standard Deviation

Equation

(IMD1 - CS1) / CS2 > 3
AND (IMD1 - CS3) / CS4 > 3

This equation checks that the standard score of the current max interval is within 3 standard deviations of historical data from last month and last year. If greater than 3 standard deviations, an exception is thrown.

Example 3: Detect unexpected zero usage (method 1)



Historical Statistics

Statistics MC Type: MC_STATS_LASTMONTH

Total Count	15 min	2880
Zero Count	15 min	3

Statistics MC Type: MC_STATS_LASTYEAR

Total Count	15 min	2880
Zero Count	15 min	0

Dynamic Values

Variable Name: IMD1
- Source Type: IMD
- Calc Function: Sum

Variable Name: CS_T1
- Source Type: Channel Statistics
- MC Type: MC_STATS_LASTMONTH
- Value: Total Count

Variable Name: CS_Z1
- Source Type: Channel Statistics
- MC Type: MC_STATS_LASTMONTH
- Value: Zero Count

Variable Name: CS_T2
- Source Type: Channel Statistics
- MC Type: MC_STATS_LASTYEAR
- Value: Total Count

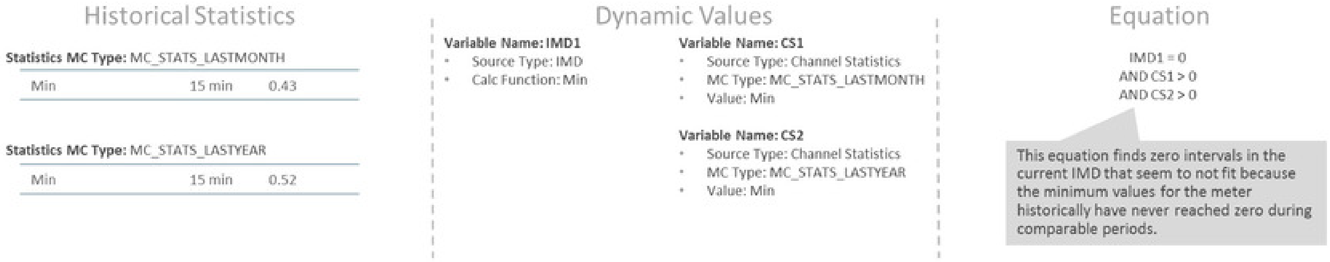
Variable Name: CS_Z2
- Source Type: Channel Statistics
- MC Type: MC_STATS_LASTYEAR
- Value: Zero Count

Equation

IMD1 = 0
AND CS_Z1 / CS_T1 < 0.01
AND CS_Z2 / CS_T2 < 0.01

This equation finds zero intervals in the current IMD that seem to not fit with historical trends for the customer based on prior counts of zero intervals.

Example 4: Detect unexpected zero usage (method 2)



Historical Statistics

Statistics MC Type: MC_STATS_LASTMONTH

Min	15 min	0.43

Statistics MC Type: MC_STATS_LASTYEAR

Min	15 min	0.52

Dynamic Values

Variable Name: IMD1

- Source Type: IMD
- Calc Function: Min

Variable Name: CS1

- Source Type: Channel Statistics
- MC Type: MC_STATS_LASTMONTH
- Value: Min

Variable Name: CS2

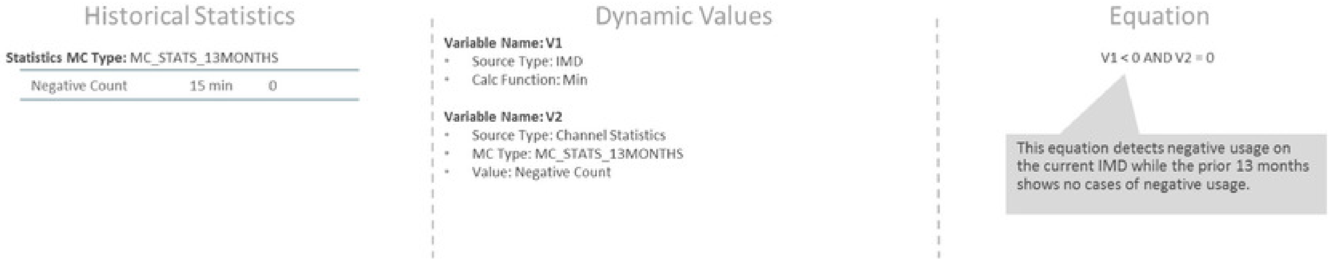
- Source Type: Channel Statistics
- MC Type: MC_STATS_LASTYEAR
- Value: Min

Equation

IMD1 = 0
AND CS1 > 0
AND CS2 > 0

This equation finds zero intervals in the current IMD that seem to not fit because the minimum values for the meter historically have never reached zero during comparable periods.

Example 5: Detect negative usage while ruling out known cases



Historical Statistics

Statistics MC Type: MC_STATS_13MONTHS

Negative Count	15 min	0

Dynamic Values

Variable Name: V1

- Source Type: IMD
- Calc Function: Min

Variable Name: V2

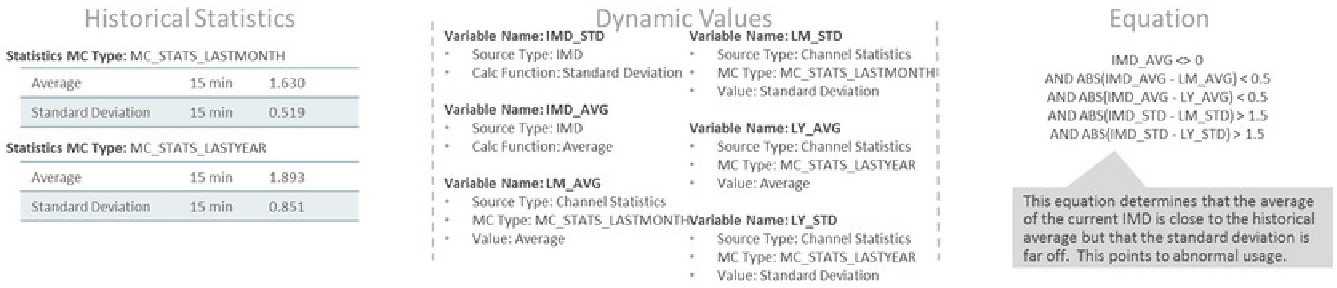
- Source Type: Channel Statistics
- MC Type: MC_STATS_13MONTHS
- Value: Negative Count

Equation

V1 < 0 AND V2 = 0

This equation detects negative usage on the current IMD while the prior 13 months shows no cases of negative usage.

Example 6: Abnormal usage



Historical Statistics

Statistics MC Type: MC_STATS_LASTMONTH

Average	15 min	1.630
Standard Deviation	15 min	0.519

Statistics MC Type: MC_STATS_LASTYEAR

Average	15 min	1.893
Standard Deviation	15 min	0.851

Dynamic Values

Variable Name: IMD_STD

- Source Type: IMD
- Calc Function: Standard Deviation

Variable Name: IMD_AVG

- Source Type: IMD
- Calc Function: Average

Variable Name: LM_AVG

- Source Type: Channel Statistics
- MC Type: MC_STATS_LASTMONTH
- Value: Average

Variable Name: LM_STD

- Source Type: Channel Statistics
- MC Type: MC_STATS_LASTMONTH
- Value: Standard Deviation

Variable Name: LY_AVG

- Source Type: Channel Statistics
- MC Type: MC_STATS_LASTYEAR
- Value: Average

Variable Name: LY_STD

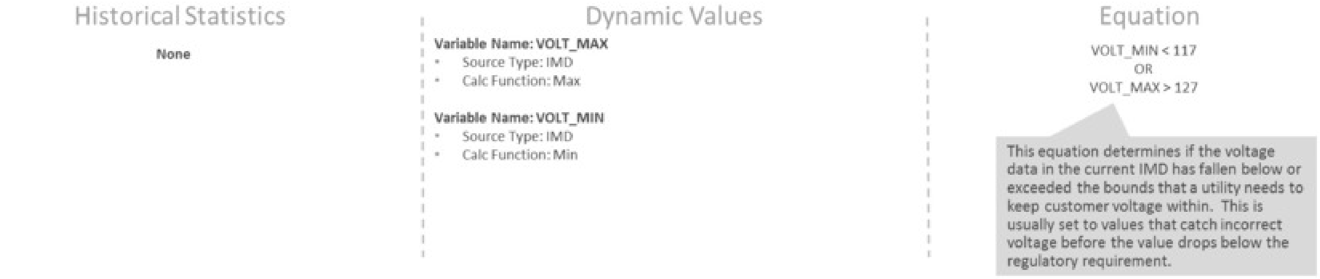
- Source Type: Channel Statistics
- MC Type: MC_STATS_LASTYEAR
- Value: Standard Deviation

Equation

IMD_AVG <> 0
AND ABS(IMD_AVG - LM_AVG) < 0.5
AND ABS(IMD_AVG - LY_AVG) < 0.5
AND ABS(IMD_STD - LM_STD) > 1.5
AND ABS(IMD_STD - LY_STD) > 1.5

This equation determines that the average of the current IMD is close to the historical average but that the standard deviation is far off. This points to abnormal usage.

Example 7: Voltage threshold monitoring



Historical Statistics

None

Dynamic Values

Variable Name: VOLT_MAX

- Source Type: IMD
- Calc Function: Max

Variable Name: VOLT_MIN

- Source Type: IMD
- Calc Function: Min

Equation

VOLT_MIN < 117
OR
VOLT_MAX > 127

This equation determines if the voltage data in the current IMD has fallen below or exceeded the bounds that a utility needs to keep customer voltage within. This is usually set to values that catch incorrect voltage before the value drops below the regulatory requirement.

Ensure IMD Exists for Sibling MCs

Overview

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

Overview

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.

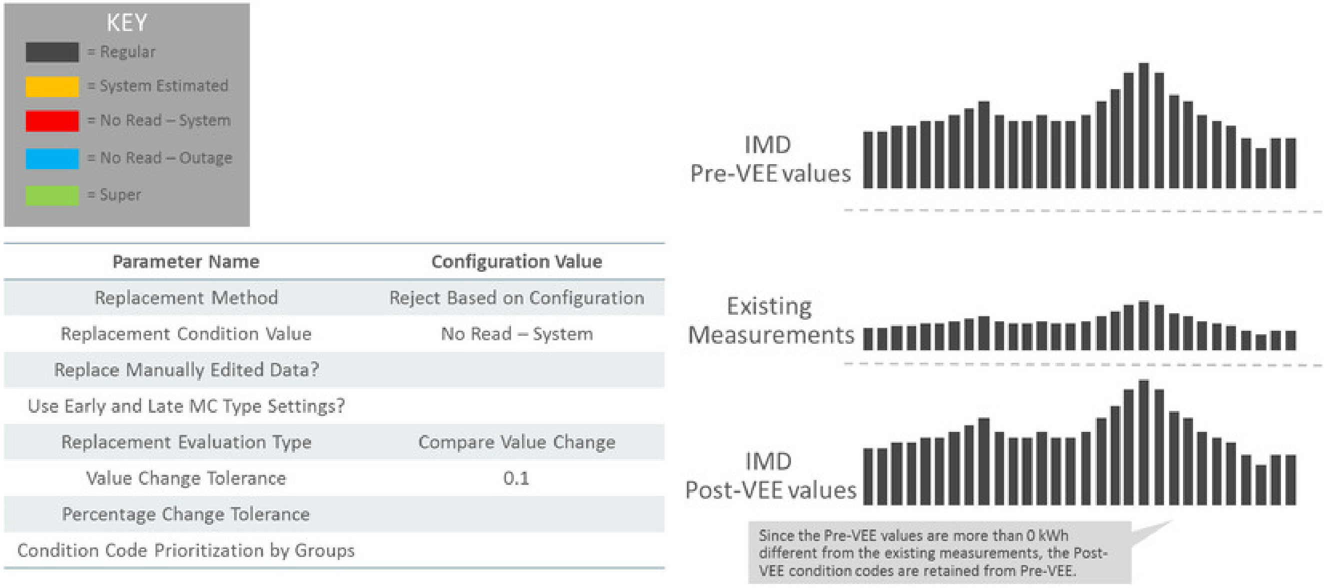
Business Object

D2-FinalMeasurementValidation

Example Scenarios

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



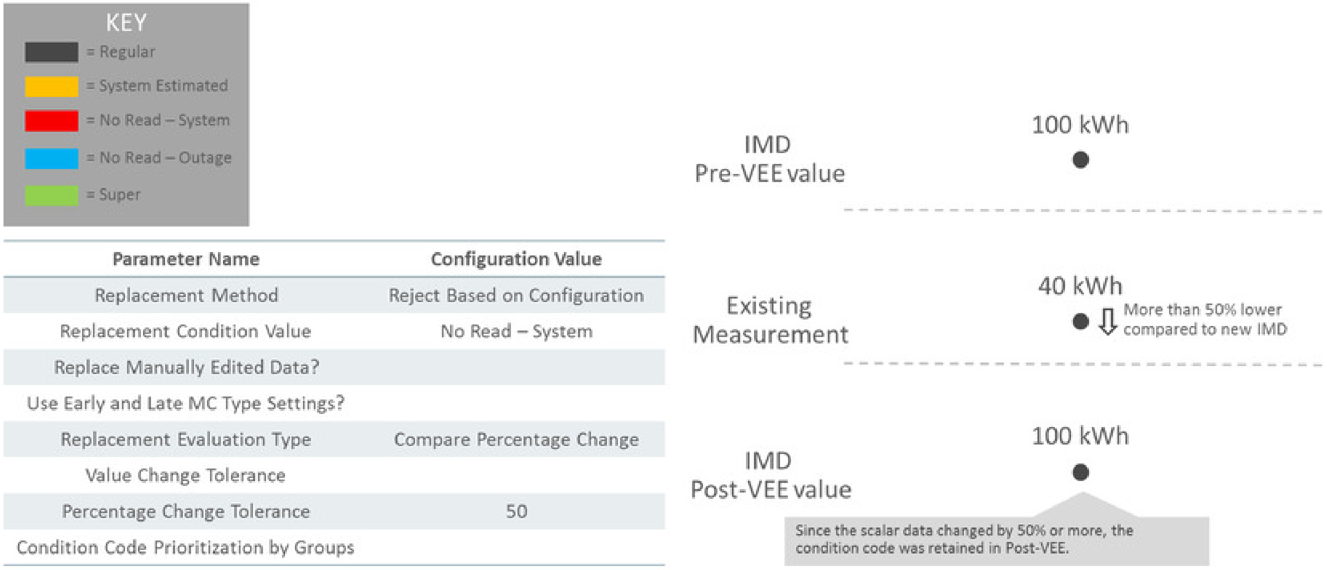
KEY

⬛	= Regular
🟨	= System Estimated
🟥	= No Read – System
🟦	= No Read – Outage
🟩	= Super

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	

Since the Pre-VEE values are more than 0 kWh different from the existing measurements, the Post-VEE condition codes are retained from Pre-VEE.

Example 2: Percentage change to replace measurement



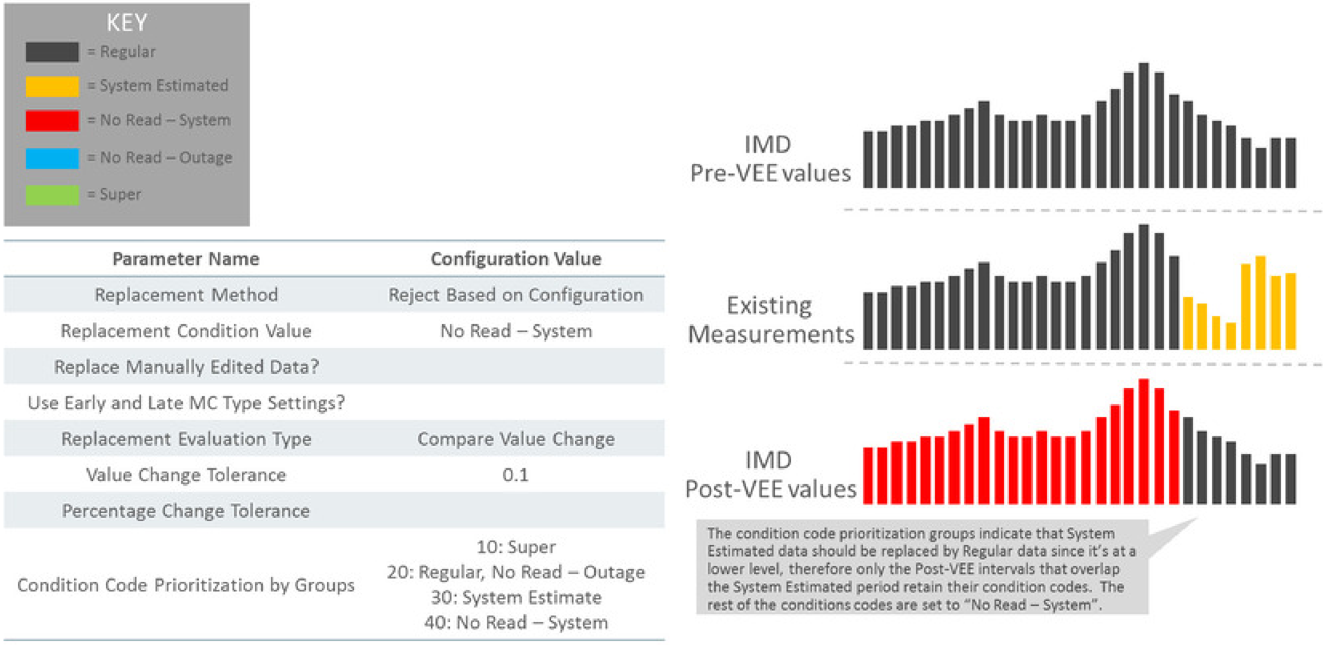
KEY

⬛	= Regular
🟨	= System Estimated
🟥	= No Read – System
🟦	= No Read – Outage
🟩	= Super

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	

Since the scalar data changed by 50% or more, the condition code was retained in Post-VEE.

Example 3: Partial replacement of estimated data with regular data based on condition range prioritization



KEY

⬛ = Regular

🟨 = System Estimated

🟥 = No Read – System

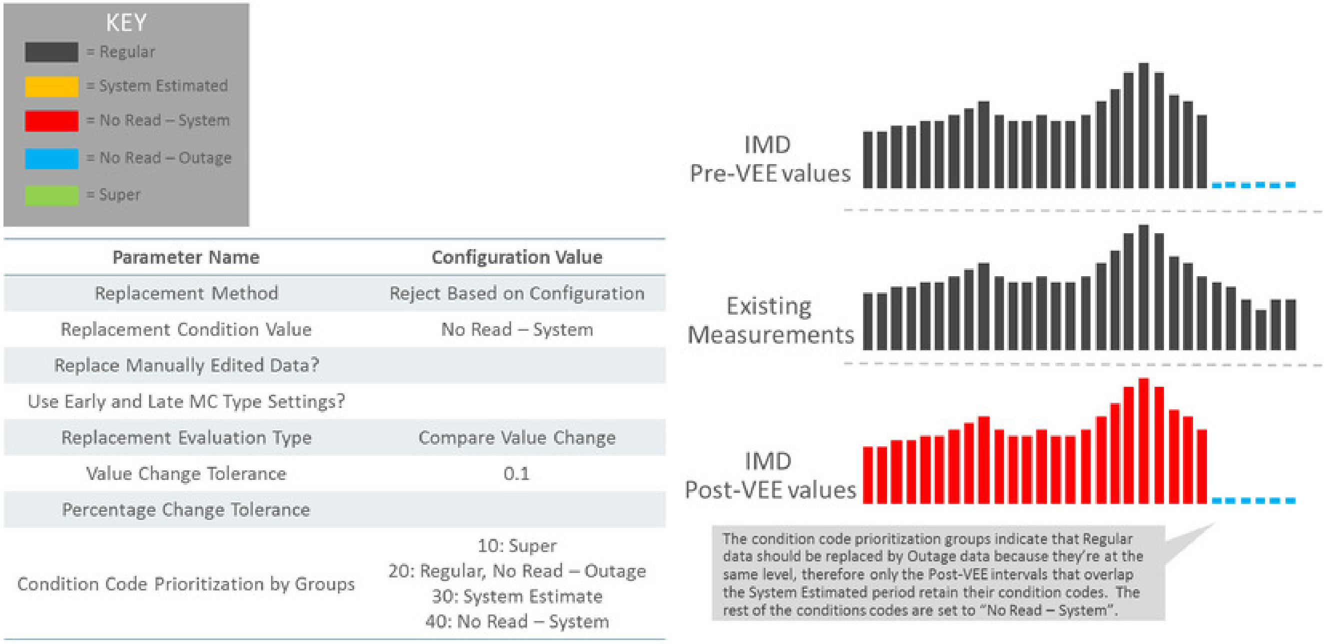
🟦 = No Read – Outage

🟩 = Super

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 20: Regular, No Read – Outage 30: System Estimate 40: No Read – System

The condition code prioritization groups indicate that System Estimated data should be replaced by Regular data since it's at a lower level, therefore only the Post-VEE intervals that overlap the System Estimated period retain their condition codes. The rest of the conditions codes are set to "No Read – System".

Example 4: Partial replacement of regular data with outage data based on condition range prioritization



KEY

⬛ = Regular

🟨 = System Estimated

🟥 = No Read – System

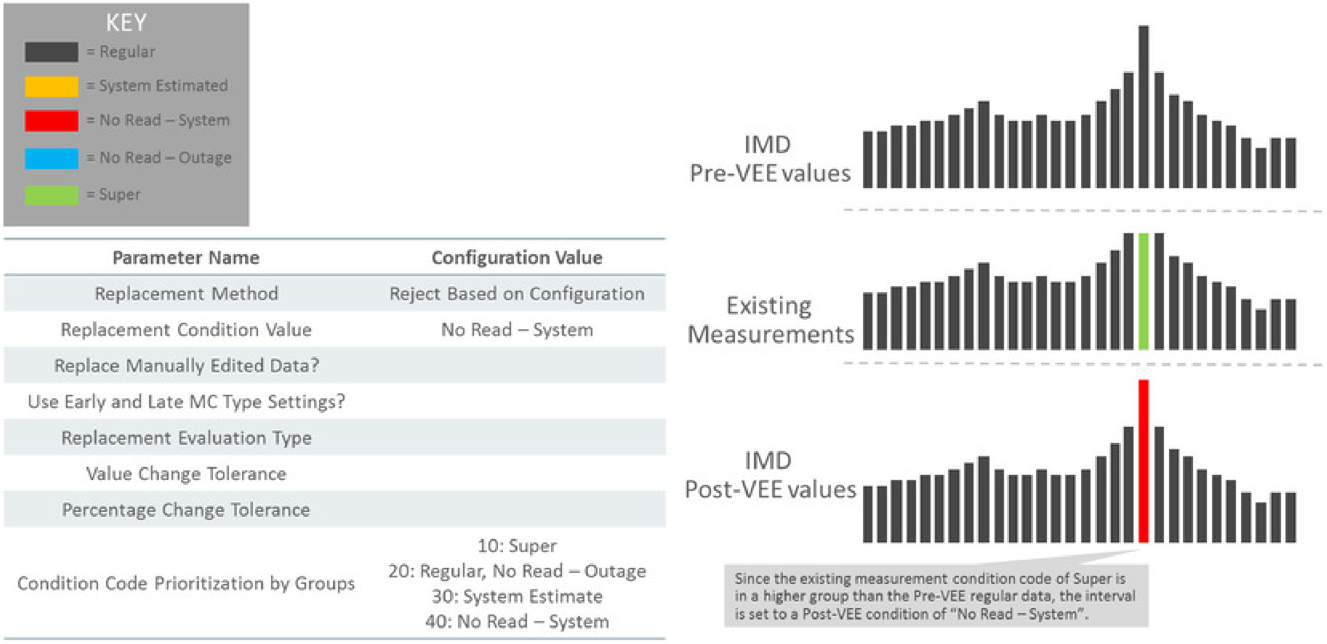
🟦 = No Read – Outage

🟩 = Super

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 20: Regular, No Read – Outage 30: System Estimate 40: No Read – System

The condition code prioritization groups indicate that Regular data should be replaced by Outage data because they're at the same level, therefore only the Post-VEE intervals that overlap the System Estimated period retain their condition codes. The rest of the conditions codes are set to "No Read – System".

Example 5: Super not replaced by regular (retain a smoothed spike)



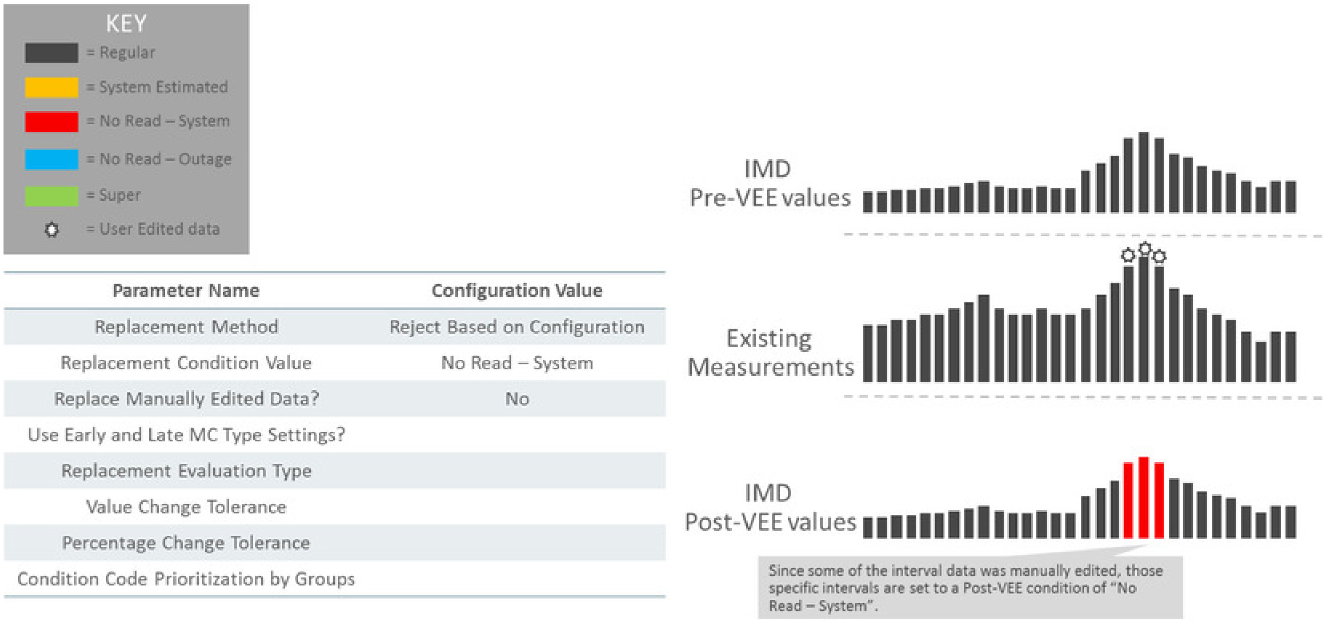
KEY

- ⬛ = Regular
- 🟨 = System Estimated
- 🟥 = No Read – System
- 🟦 = No Read – Outage
- 🟩 = Super

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 20: Regular, No Read – Outage 30: System Estimate 40: No Read – System

Since the existing measurement condition code of Super is in a higher group than the Pre-VEE regular data, the interval is set to a Post-VEE condition of "No Read – System".

Example 6: Prevent replacement of manually edited data



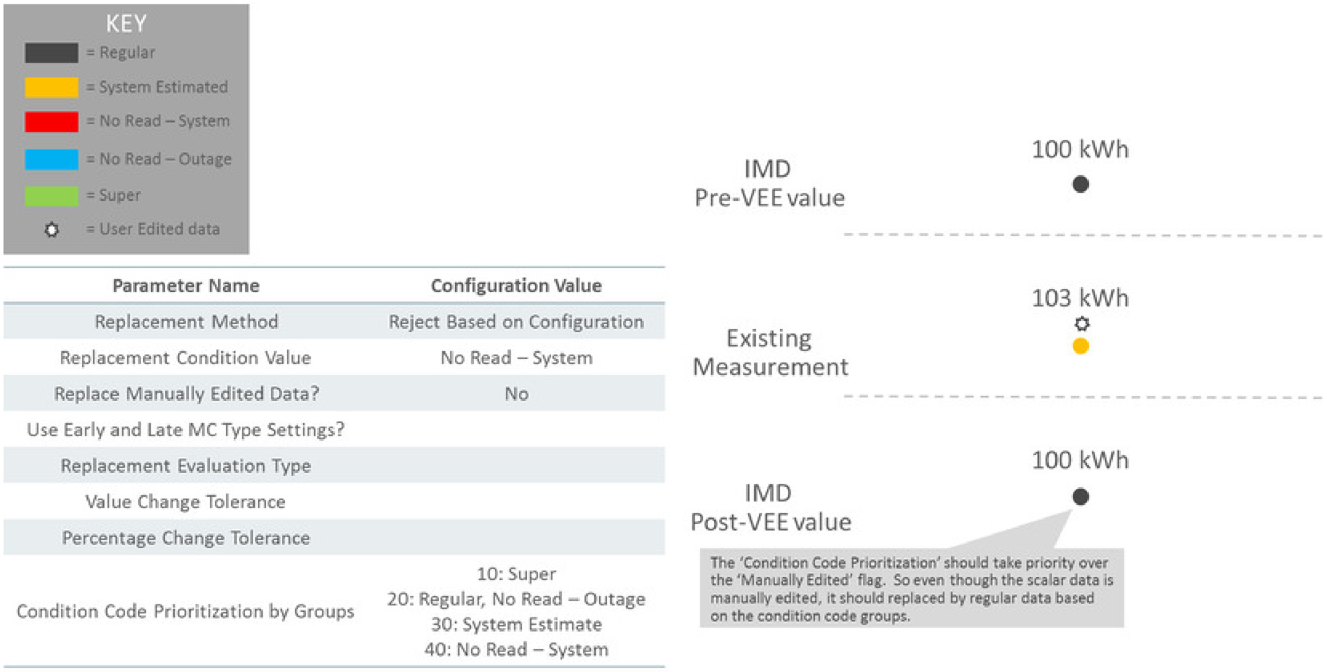
KEY

- ⬛ = Regular
- 🟨 = System Estimated
- 🟥 = No Read – System
- 🟦 = No Read – Outage
- 🟩 = Super
- ⚙ = User 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	

Since some of the interval data was manually edited, those specific intervals are set to a Post-VEE condition of "No Read – System".

Example 7: Allow replacement of manually edited data if the condition group is lower than new 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	10: Super 20: Regular, No Read – Outage 30: System Estimate 40: No Read – System

The 'Condition Code Prioritization' should take priority over the 'Manually Edited' flag. So even though the scalar data is manually edited, it should replaced by regular data based on the condition code groups.

High/Low Check

Overview

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
- As well as other important configuration

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

Overview

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.

- 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.

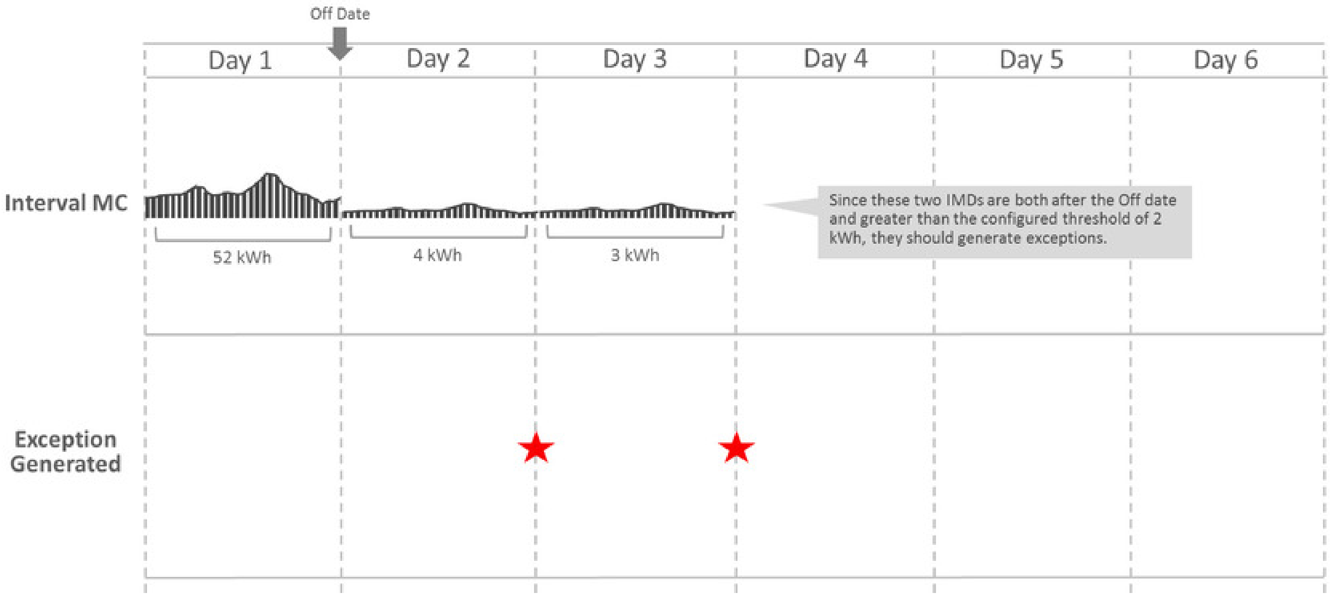
Business Object

D2-InactiveMeasurementCheck

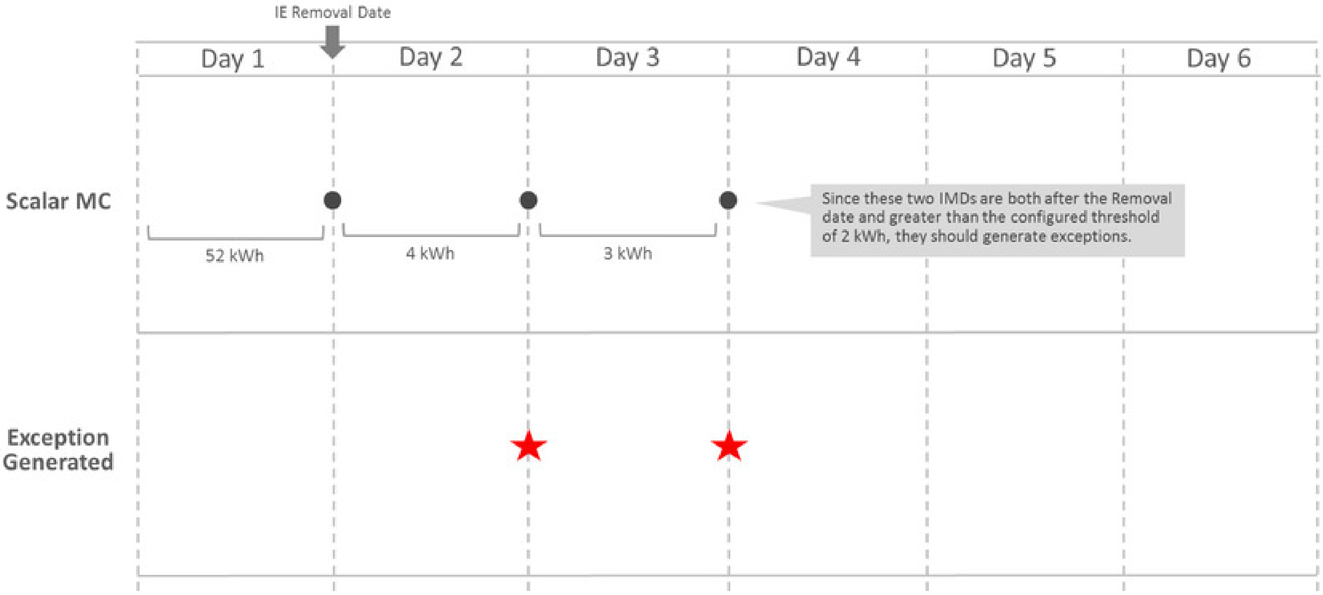
Example Scenarios

Below are some example scenarios that can be achieved based on configuration of this rule.

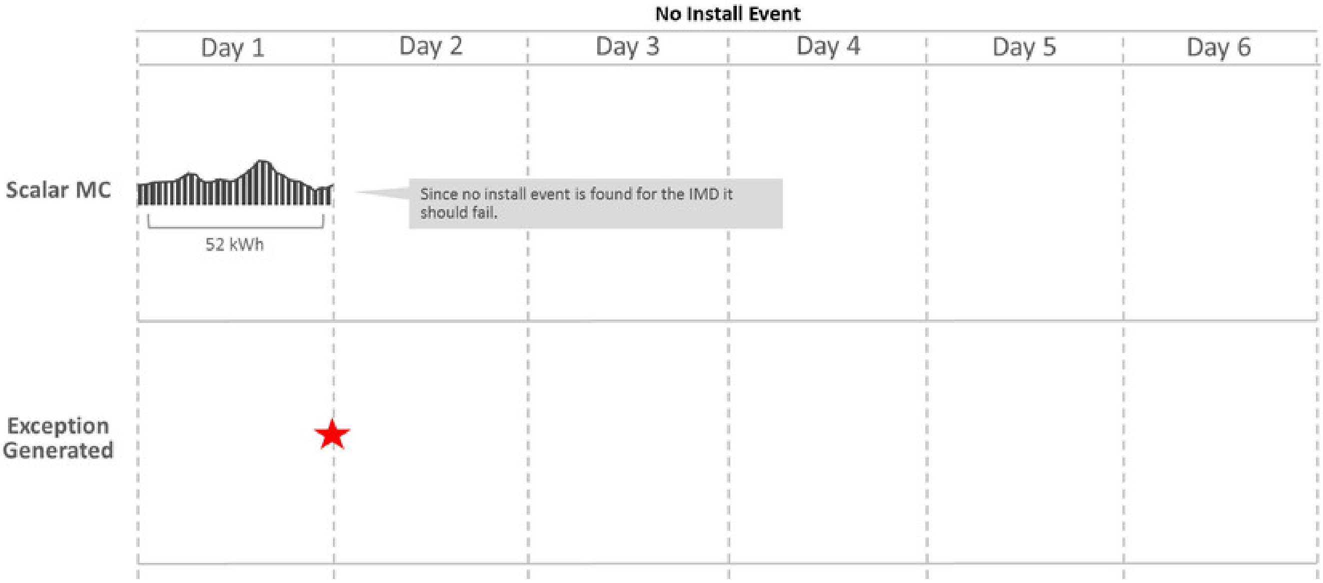
Example 1: Non-zero data during disconnected period


Off Date
Day 1
Day 2
Day 3
Day 4
Day 5
Day 6
Interval MC
52 kWh
4 kWh
3 kWh
Since these two IMDs are both after the Off date and greater than the configured threshold of 2 kWh, they should generate exceptions.
Exception Generated

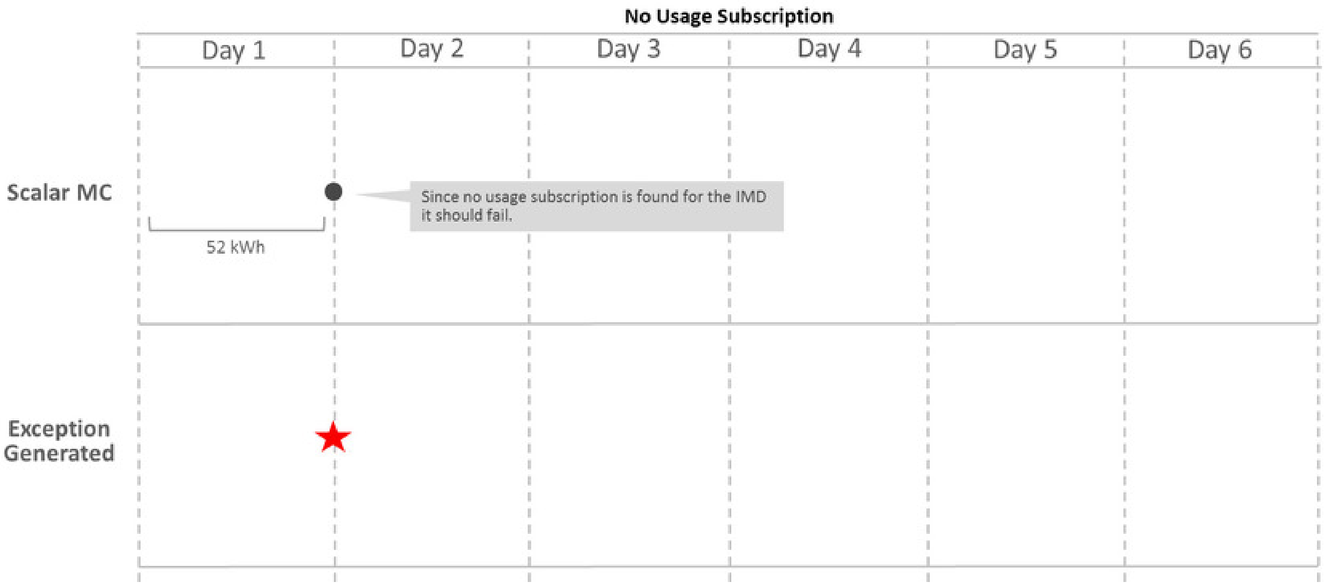
Example 2: Non-zero data after removal date with no new install


IE Removal Date
Day 1
Day 2
Day 3
Day 4
Day 5
Day 6
Scalar MC
52 kWh
4 kWh
3 kWh
Since these two IMDs are both after the Removal date and greater than the configured threshold of 2 kWh, they should generate exceptions.
Exception Generated

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

Overview

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

Overview

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

Overview

This rule validates that the register multiplier supplied with the Initial Measurement is equal to the multiplier stored on the measuring component.

➤ **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

Overview

This rule checks if the total consumption of the IMD is less than zero. If the rule encounters negative consumption, an exception will be logged only if the related Measuring Component Type is not configured to "allow negative consumption".

➤ **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

Overview

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.

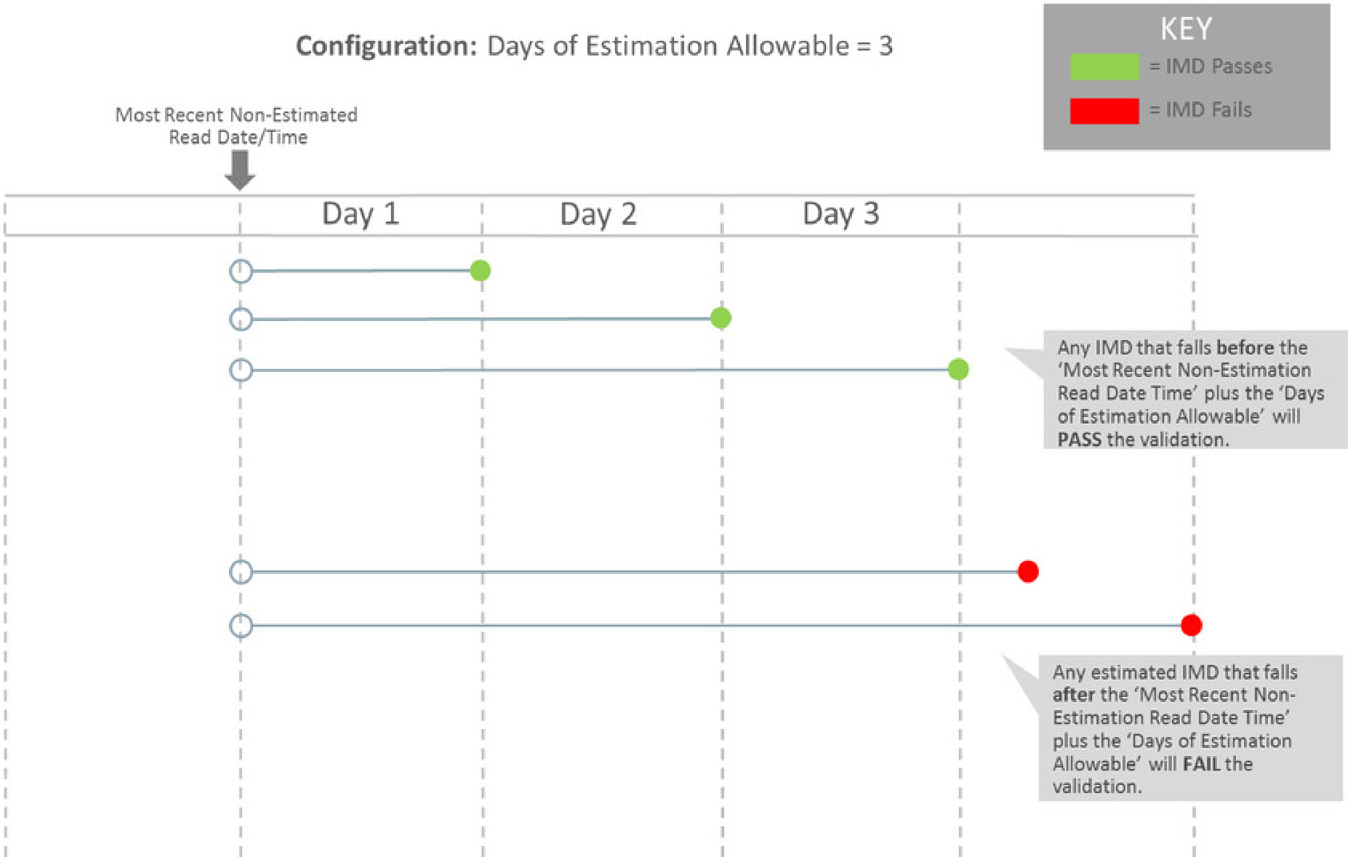
For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

Business Object

D2-ProlongedEstimationCheck

Example Scenarios

Below are some example scenarios that can be achieved based on configuration of this rule.


Configuration: Days of Estimation Allowable = 3
KEY
🟩 = IMD Passes
🟥 = IMD Fails
Most Recent Non-Estimated Read Date/Time
Day 1
Day 2
Day 3

Any IMD that falls **before** the 'Most Recent Non-Estimation Read Date Time' plus the 'Days of Estimation Allowable' will **PASS** the validation.

Any estimated IMD that falls **after** the 'Most Recent Non-Estimation Read Date Time' plus the 'Days of Estimation Allowable' will **FAIL** the validation.

Raise Missing Quantity Exception

Overview

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

Overview

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.

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

Overview

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

Overview

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.

Business Object

D2-ZeroConsumptionCheck

Estimation VEE Rules

Interval Adjustment From Scalar

Overview

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

Interval Averaging Estimation

Overview

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

Overview

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.

> **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.

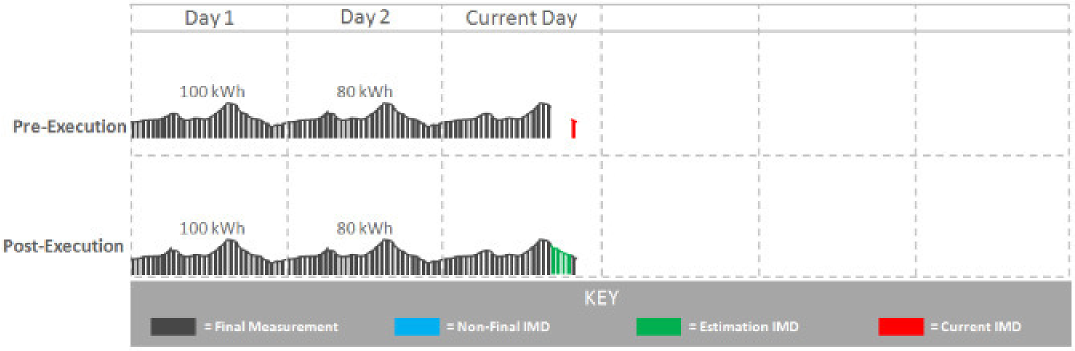
Business Object

D2-CreateEstimationIMDRule

Example Scenarios

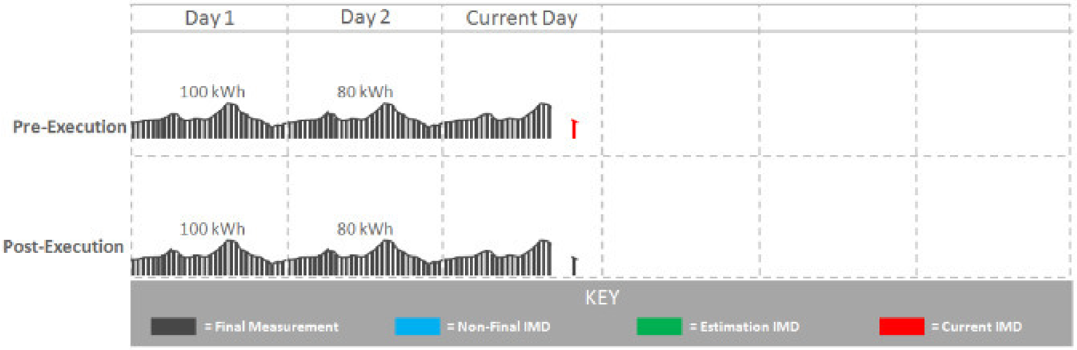
The following are sample scenarios that can be achieved based on configuration of this rule.

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".



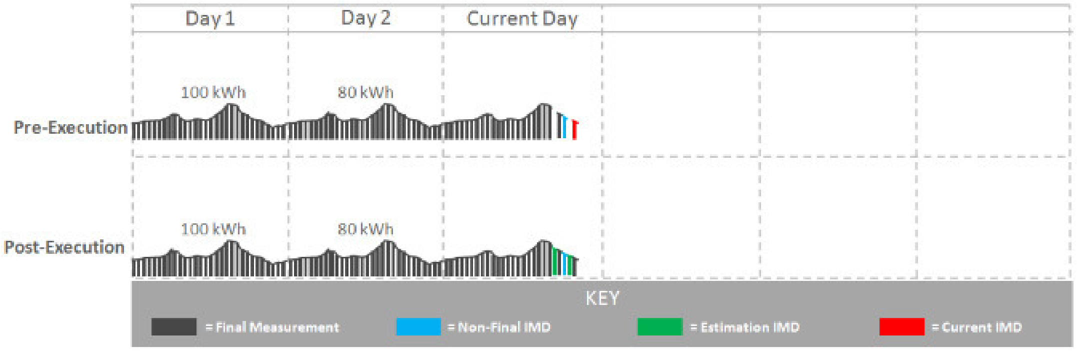
Since the maximum hours has not been exceeded and the rule is configured to always fill the gap, a 4-hour estimation IMD is created to fill the gap.

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.

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

Overview

This algorithm 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

D2-IntervalInterpolationEst

Interval Profile Estimation

Overview

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

Scalar Calculation From Interval

Overview

This algorithm 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 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

Overview

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

Overview

This VEE Rule algorithm 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

Overview

This rule will prorate 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.

➤ **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

Overview

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

Overview

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

Overview

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

Overview

This rule allows VEE to be successfully terminated based on a list of exceptions.

> **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)

Overview

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

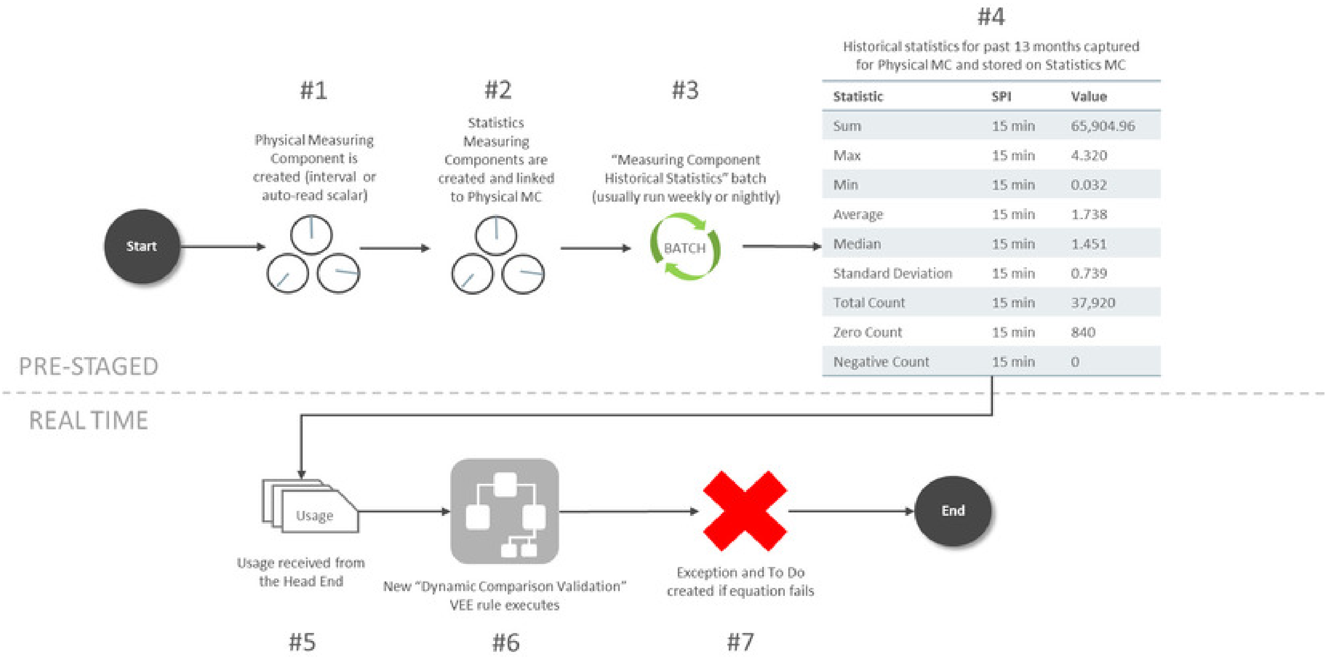
Understanding Measuring Component Statistics

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:


#1
Physical Measuring Component is created (interval or auto-read scalar)
#2
Statistics Measuring Components are created and linked to Physical MC
#3
"Measuring Component Historical Statistics" batch (usually run weekly or nightly)
BATCH
#4
Historical statistics for past 13 months captured for Physical MC and stored on Statistics MC
Start
PRE-STAGED
REAL TIME
Usage
Usage received from the Head End
#5
New "Dynamic Comparison Validation" VEE rule executes
#6
Exception and To Do created if equation fails
#7
End

Statistic	SPI	Value
Sum	15 min	65,904.96
Max	15 min	4.320
Min	15 min	0.032
Average	15 min	1.738
Median	15 min	1.451
Standard Deviation	15 min	0.739
Total Count	15 min	37,920
Zero Count	15 min	840
Negative Count	15 min	0

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:

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 *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 *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 *This increases the Measuring Component table by an extra 25% to 75% from its size without statistics.*
Additional Measurements *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 *This data is a minuscule fraction compared to the overall AMI data footprint.*
Additional Measurements *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) *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:
- Period End Lag Days = 0
- Period Start Lag Days = 30
- "Same Month Last Year" Channel Statistics Type:
- Period End Lag Days = 365
- Period Start Lag Days = 395
- "Prior 13 Months" Channel Statistics Type:
- Period End Lag Days = 0
- Period Start Lag 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

This section describes concepts and common tasks related to usage administration.

The Big Picture of Usage Requests

Oracle Utilities Meter Data Management is used to Some organizations use a meter data management (mdm) system to record meter reading information. Since meter reads are not available in Oracle Utilities Customer Care and Billing to calculate consumption during the billing process, usage (or bill determinants) must be requested by Oracle Utilities Customer Care and Billing from the mdm system, Oracle Utilities Meter Data Management

The following sections describe how the base-package usage request process works.

Requesting Bill Determinants

The term bill determinant request is another way to refer to a usage request. These usage requests are created during the batch billing process, and also when a user generates an online bill segment or performs cancel / rebill. The get consumption algorithm specified on the SA Type's Bill Segment Type is responsible for creating the usage request.

Online Billing Usage Requests

- Usage requests created from an online billing request are not held in the initial state. These usage requests are transitioned through their lifecycle resulting in the request being sent to Oracle Utilities Meter Data Management (MDM) and, once a response is received, the bill segment is regenerated.
- The bill segment remains in the Freezable state until the user freezes the bill segment and completes the bill. If the freezable bill segment is regenerated, however, the system cancels the usage request and creates another usage request for the bill segment.
- In addition to the standard MDM usage request elements, the system captures the responsible user id.

Usage Request Lifecycle

The usage request business object (BO) contains the rules that govern the processing of a usage request. The base product provides the BO C1-UsageRequest which serves as a parent BO and contains the following lifecycle:

- All usage requests are created in the initial Pending state. If the usage request was created from the batch billing process, the batch Usage Scheduled Monitor Process must be executed to transition the usage request.
- If there are pending sync requests for the usage request's service agreement, the usage request is held in the Awaiting Data Sync state until the sync is processed.

- Once all pending sync requests related to the usage request have been processed, the usage request transitions to the Send Request state. An enter algorithm on this transitory state is responsible for sending the usage request to Oracle Utilities Meter Data Management (MDM).
- The usage request sits in the Awaiting Bill Determinants state until a response is received from MDM, or a time out is encountered.
- When a response is received from MDM, the usage request is transitioned to either the Bill Determinants Received state, or the Error state. In the case of an error, the following occurs:
  - If the usage request was created from the batch billing process and the billing window is still open, a new usage request will be created the next time billing runs and the process is repeated. If however, the billing window is no longer open, a to do entry is created for manual follow up.
  - If the usage request was created from an online billing process, the user will be notified. The bill segment remains in the error state and the user can either delete the bill segment, or attempt to generate it again. If the bill segment is regenerated, a new usage request is created and the process is repeated.
  - The system transitions usage requests in the Bill Determinants Received state to the Bill Determinants Processed state. An enter algorithm on this final state is responsible for generating the bill segment. If the usage request was created from the batch billing process, the bill segment is frozen and the bill is completed. For online billing usage requests, the user is responsible for freezing the bill segment and completing the bill.

Summary Billed Accounts and Usage Requests

A "summary" account refers to an account with service agreements that cover more than one premise. If these premises are linked to service points that span different measurement cycle routes scheduled for reading on different dates, there's no guarantee that all the account's SAs will be billed at the same time. This is because batch billing for manually read meters is based on the meter's scheduled reading date.

For example, assume a summary account has two SAs each linked to a different premise. One is on route X, scheduled for reading every first of the month and the other is on route Y, scheduled for reading every 20th of the month. Assume that the account’s bill cycle is cycle X (which matches the schedule of premise/SP on route X). When billing runs on the first of the month, the service agreements tied to route Y won’t have any reads. And because route Y is not scheduled for reading, the system will not estimate the consumption. Instead, it will skip billing the route Y SAs until the next month. This means some SAs are always one billing cycle behind.

Requesting Bill Determinants for Summary Billed Accounts

Usage requests for summary billed accounts are handled as follows:

- On the first night of the billing window usage requests are created for each SA and sent to Oracle Utilities Meter Data Management (MDM). MDM then checks what the next schedule read date is for each service point.
- Assuming Oracle Utilities Meter Data Management owns meter read schedules; when batch billing executes Oracle Utilities Customer Care and Billing (CCB) cannot determine whether a summary account’s service agreement should be included on the bill or not.
- If MDM determines that an SA/usage subscription should be skipped from billing, it notifies CCB by including a skip indicator and reason on the usage response, as well as the SA’s next scheduled read date. This date is captured as a ***MDM Next Scheduled Read Date*** characteristic on the SA, and the usage request is cancelled. The system still attempts to complete the bill since this might be the last outstanding usage request for the bill
- When billing next runs, the base ***Get bill segment consumption using a usage request*** algorithm first checks if the SA should be skipped as follows (Note that this check is also performed in first billing run):
- Get the ***MDM Next Scheduled Read Date*** characteristic on the SA. If this date is after the bill segment end date, we’ll skip the SA from billing

If your implementation would like to wait until the bill cycle window end date before attempting to complete summary account’s bills, ensure that the ***Complete Bill at End of Bill Cycle*** pre-bill completion algorithm is defined on customer class. This is useful if scheduled reads are expected in the bill cycle window, but not necessarily on the window start date.

Corrected Read Notifications

If a read that was used for bill determinant calculations is modified in the mdm system, notification is sent to Oracle Utilities Meter Data Management.. Resulting in the creation of an off cycle bill generator. The system uses the business object defined as the OCBG Corrected Read BO MO option on the Off Cycle Bill Generator maintenance object to create the OCBG.

What happens next depends on the lifecycle that your implementation has configured for the OCBG Corrected Read BO. Here are examples of what might occur:

- Create a To Do entry for manual follow up.
- Find any frozen bill segments that might be affected by the corrected read and perform cancel/rebill.

Configuring The System For Usage Request Integration

Oracle Utilities Customer Care and Billing sends usage requests to Oracle Utilities Meter Data Management (MDM) in the form of an xml message. MDM sends responses back to Oracle Utilities Customer Care and Billing.

The following sections describe at a high level the data setup required to send usage requests to MDM.

Start and End Times for Billing

As you know, there is logic in billing to determine the start date and end date for a bill segment. Refer to Ways To Control The Start Date Of A Bill Segment and Ways To Control The End Date Of A Bill Segment for more information. When billing for a customer with interval data, the system also needs to know the time.

The time used by billing, referred to as the cutoff time, is stored on the service agreement. There is also a control on the service agreement called Start Day Option that determines which day to use for the start time. Billing uses the billing date, the cutoff time, and the start day option to determine the correct interval data to process.

Oracle Utilities Customer Care and Billing lacks knowledge of the type of meter installed at a service point. Thus, both interval processing period as well as scalar processing information is captured on a usage request. Oracle Utilities Meter Data Management (MDM) then uses the appropriate period to calculate bill determinants based on the type of meter installed.

MDM returns the true usage period used to calculate bill determinants on the usage response. This usage period is captured on the bill segment.

The Big Picture of Time of Use Mapping and Pricing

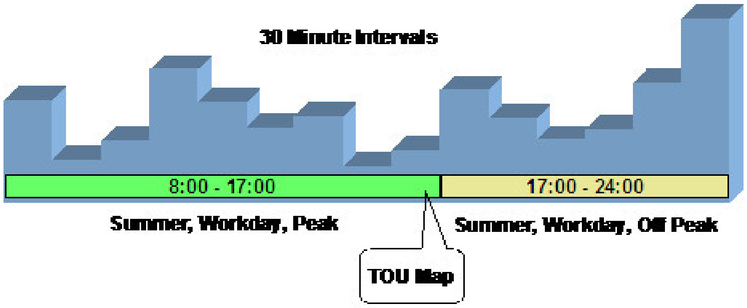
This section provides an overview of the concepts related to setting up your control tables to support time of use mapping and pricing with the rates.

NOTE: Time of use mapping is supported to enable low volume interval data manipulation (for example, for mapping electric vehicle charging events into time of use quantities). A meter data management application, such as Oracle Utilities Meter Data Management, must be used to manipulate high volumes of interval data and calculating bill determinants for billing purposes.

Mapping to Time of Use (TOU) Periods

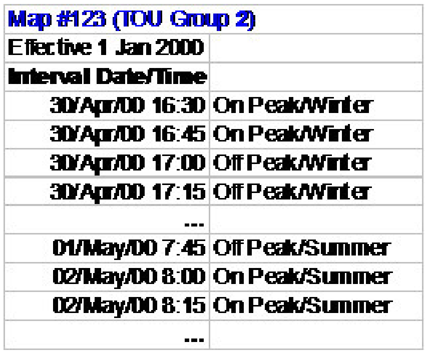
Many customers choose not to price their interval data using interval prices. Customers may choose for their interval data to be mapped into time of use (TOU) periods. This option for interval data might be preferred because:

- Typically it involves fixed prices for the use periods
- It is more manageable than direct billing
- It is easier for a customer to forecast and budget



A TOU map's purpose is to define the TOU codes for a collection of time period definitions (i.e. given dates and times). The TOU Map has a TOU Map Type, which defines the interval size between TOU map data rows.

Time of use periods can (and often do) change during the year.



Map #123 (TOU Group 2)	
Effective 1 Jan 2000	
Interval Date/Time	
30/Apr/00 16:30	On Peak/Winter
30/Apr/00 16:45	On Peak/Winter
30/Apr/00 17:00	Off Peak/Winter
30/Apr/00 17:15	Off Peak/Winter

01/May/00 7:45	Off Peak/Summer
02/May/00 8:00	On Peak/Summer
02/May/00 8:15	On Peak/Summer

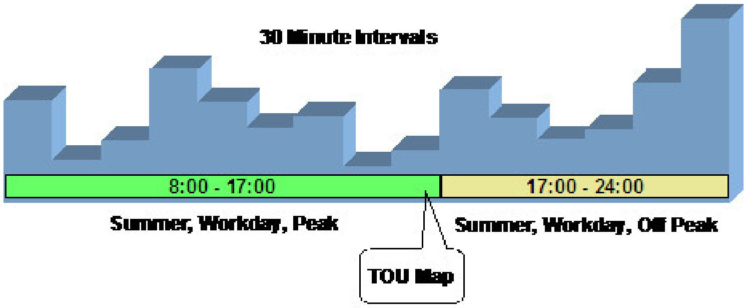
All the possible TOU codes for a given map are grouped together in a TOU group. Refer to Grouping of TOU Codes for TOU Mapping for more information.

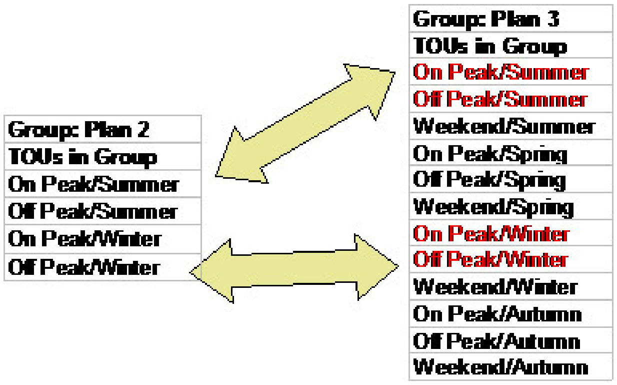
Grouping of TOU Codes for TOU Mapping

This section describes the relationship between TOU Codes and TOU Mapping.

Overview of TOU Codes and TOU Mapping

A TOU map's purpose is to define the TOU codes for a collection of time period definitions (i.e. given dates and times).





Group: Plan 2
TOUs in Group
On Peak/Summer
Off Peak/Summer
On Peak/Winter
Off Peak/Winter

Group: Plan 3
TOUs in Group
On Peak/Summer
Off Peak/Summer
Weekend/Summer
On Peak/Spring
Off Peak/Spring
Weekend/Spring
On Peak/Winter
Off Peak/Winter
Weekend/Winter
On Peak/Autumn
Off Peak/Autumn
Weekend/Autumn

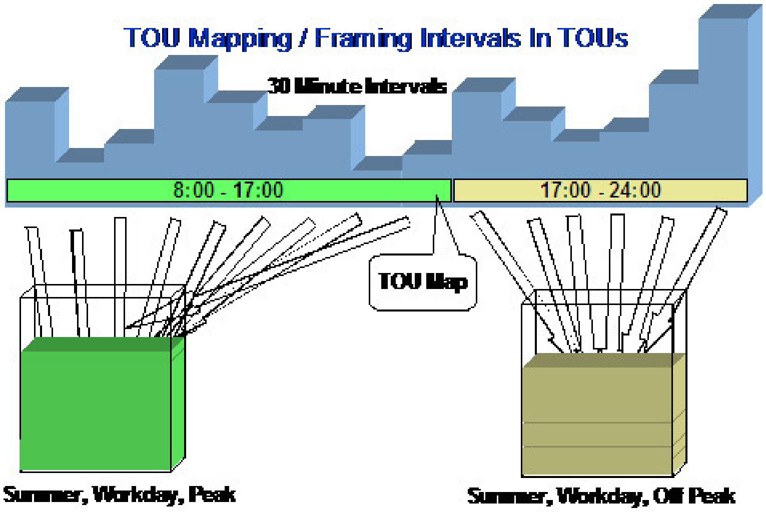
TOU Sequence for TOU Mapping and Pricing

If desired, you may use sequence number to indicate the relative position or relative priority of each TOU code within a TOU group. This sequence number is not used by any system functionality, but is available for you to use in a plug-in algorithm.

Time of Use (TOU) Mapping and Pricing

The following section describes the logic used by the system to map an interval data curve to TOU periods using a TOU map, and subsequently apply prices to these mapped quantities.

At some point during billing, the system will take an interval data curve and will map the interval values to TOU periods based on a TOU Map. The interval data curve is retrieved from either a specific usage request or from the new rate engine buffer populated prior to calling the rate engine.



This type of data manipulation is typically handled using a calculation rule based on the 'math' calculation rule type.

Once the interval data curve values have been mapped to TOU periods, prices can be applied to the time of use period quantities using calculation rules based on the 'service quantity' calculation rule type.

Calculation Rules for TOU Mapping & Pricing

Calculation Rules are used to perform time of use mapping and time of use pricing.

Time of Use Mapping

Time of use mapping will be typically handled using a calculation rule based on the 'math' calculation rule type. Calculation rules of this type can perform many functions. Two examples include:

- Applying a TOU map to an interval data curve to produce service quantities that will be added to the bill segment's SQ collection.
- Applying a TOU map to a derived interval data curve. For example, after deriving the power factor curve, perform TOU mapping on the resultant curve.

For further details on the 'math' calculation rule type, refer to Base Package Math Calculation Rule for more information.

Time of Use Pricing

Time of use pricing will be typically handled using calculation rules based on the 'service quantity' calculation rule type. Calculation rules of this type can perform many functions. For example:

- Add or update a service quantity in the bill segments SQ collection
- Create bill calculations lines that levy charges based on some type of consumption. For example, applying a price to a specific service quantity in the bill segment's SQ collection

For further details on the 'service quantity' calculation rule type, refer to Base Package Service Quantity Calculation Rule for further information.

NOTE: Calculation rules are implemented using algorithms that contain the logic to be performed. If your implementation requires a new type of calculation rule, you can configure a new calculation rule type and reference the appropriate algorithm on it.

TOU Map Used For Mapping

An interval data curve is mapped into time-of-use periods using a TOU map. The TOU map used by the calculation rule for mapping interval data curve values to time of use periods is defined directly on the calculation rule.

Attributes of TOU Maps

TOU map types define attributes shared by TOU maps of a given type. The essential attributes of any TOU Map Type are the interval sizes between TOU map data rows, associated TOU map template(s) to use, and whether the TOU data will follow any seasonal time shifting.

Designing Your Time of Use (TOU) Mapping and Pricing Options

This section provides an overview for designing your control tables to support time of use (TOU) mapping and pricing with the new rating engine.

Designing Your Time Of Use (TOU) Mapping and Pricing Calculation Rules

As you know from the rates chapter, the system can handle mapping of interval data curve values into different time of use periods and pricing these quantities using the 'math' and 'service quantity' calculation rule types respectively. Refer to Base Package Calculation Rule Descriptions for more information.

The following guidelines provides a high level outline on how you should go about designing a rate schedule that involves time of use mapping and pricing.

- Obtain copies of existing bills that use the rate in question. If the rate is new, then write up exactly how the information should appear on the customers' printed bills.
- Identify all the lines that represent charges for individual time of use periods.
- Determine how the quantities for the time of use periods are calculated. Which time of use map is used to define the time periods?
- In order to perform time of use mapping and pricing, calculation rules need to be designed
  - A calculation rule based on the 'math' calculation rule type may be used to map an interval data curve into time of use periods. This calculation rule must know the TOU map to apply to the interval data curve to map into the relevant time of use periods. The time of use period quantities are populated in the bill segments SQ collection.
  - Calculation rules based on the 'service quantity' calculation rule type may be used to apply prices to each mapped time of use period quantity. These calculation rules each require either a price or regular bill factor for pricing each time of use period quantity. The bill factor may contain the price directly, or may indicate that the price is customer specific and can be found as contract quantities for the service agreement.

Once you have your calculation rules designed, you will be able to design the other control tables needed to set up your time of use billing customer.

Designing Your TOU Codes for TOU Mapping

The next most logical step in designing your time of use mapping controls is to define your time of use codes. To do this, look at the time of use periods to which your usage needs to be mapped. These values will likely correspond to the time of use quantities that your rate bills for. It should be noted that it's possible that will not bill for every time of use period.

Designing Your TOU Groups for TOU Mapping

To further aid in designing time of use mapping, the TOU Group enables you to group together all the time of use codes that are available to be used in a single map. You must also decide if you want to use a sequence number to define the relative order of a TOU code within a TOU group.

When TOU data is created for a TOU map, only TOU periods defined on a specified TOU group can be specified.

Designing Your TOU Map Types for TOU Mapping

Now that you have your TOU groups defined, you can begin defining TOU map types. Recall that the TOU map type defines the attributes that may be shared by TOU maps of the same type.

The essential attributes of any TOU Map Type are the interval sizes between TOU map data rows, associated TOU map template(s) to use, and whether the TOU data will follow any seasonal time shifting. Refer to Designing Your Time Options for more information.

TOU Map Type Interval Size

The interval size of a TOU map must divide evenly into the interval size of the interval data curve that uses the map (because the system joins the date/time of the interval data curve values 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 interval data curve. However, it is not OK to have a 60 minute TOU map used with a 15 minute interval data curve because the join will miss 3 out of 4 interval data curve values.

TOU Map Type Override Template

While most TOU maps will use the TOU map template defined on the TOU map type, TOU maps also support override templates. Refer to Designing Your TOU Map Templates for TOU Mapping for more information.

- A TOU map's TOU map type defines the default 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 default template.
- An individual TOU map can have override templates. If the TOU map doesn't have an override template, the default template defined on the TOU map type is used to generate the map's TOU data.

Refer to Creating TOU Map Types for further information.

Designing Your TOU Map Templates for TOU Mapping

In order to help your users to create and maintain data for TOU maps, you may define TOU map templates, which can be used to generate data for a TOU map. The templates may be used to define standard data for a TOU map as well as data for special periods, such as holidays. TOU map types reference TOU 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 in the template.

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 by specifying the following:

- A Work Calendar that defines when holidays start and end
- Either:
  - A Holiday TOU period for consumption on holiday
  - A Holiday TOU Map Template that defines the TOU codes to use for different times in the year

TOU Map Template Interval Size

TOU map templates can also specify an interval size. 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 interval size is divisible by the template's interval size. For example, a 60 minute template can be used to generate TOU data for TOU maps with an interval size of 60 minutes, 15 minutes, 5 minutes, etc. This means separate map templates are not needed for every interval size.

Refer to Creating TOU Map Templates for further information.

Setting Up Time of Use (TOU) Mapping and Pricing Control Tables

This section provides an overview of setting up your control tables to support time of use mapping and pricing with rates.

Creating TOU Map Templates

TOU map templates may be used to define standard data for a TOU map as well as data for special periods, such as holidays. TOU map types reference TOU map templates and used for TOU map data generation.

Prerequisites: You must define TOU groups and work calendars before you can create TOU map templates. Refer to the Oracle Utilities Application Framework online help for more information about creating work calendars.

To maintain existing TOU map templates, select **Admin** > **Usage** > **TOU Map Template** > **Search** then use standard actions to edit, duplicate, or delete a TOU map template.

To define a new TOU map template, follow these steps:

1. Select **Admin** > **Consumption** > **TOU Map Template** > **Add.** If your system supports more than one TOU map template business object, you will be prompted to select a business object for this TOU map template.
2. Enter a name and a meaningful description for the TOU map template.
3. Select the TOU group to be used by the TOU map template.
4. Select the default TOU for the TOU 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.
5. Select the work calendar for the TOU map template. Work calendars define the days of the week on which work is performed, and specify holidays.
6. Select the holiday TOU for the TOU map template (from the TOU Group).

7. Select the TOU map template used for holidays (if applicable).
8. Specify the interval size for TOU map data created from the map template. Interval size is designated as hours:minutes:seconds (HH:MM:SS)
9. To specify TOU schedule's date ranges and which TOUs should be used for this TOU map template, click the + or - sign in the TOU Schedule Section and enter or select the following:
    - Start and End Dates for a specific date range
    - Start and End Days of the Week: To add or remove Start and End Days of the Week pairs, click the + or - sign and select the appropriate weekdays
    - Start and End Times: To add or remove Start and End Times within a Start and End Days of the Week pair, click the + or - sign and enter the appropriate times
    - TOU
10. Click **Save.**

Now you can use the TOU map template when creating TOU types.

Creating TOU Map Types

An interval data curve is mapped into time-of-use periods using a TOU map. The TOU map used by the calculation rule for mapping interval data curve values to time of use periods is defined directly on the calculation rule. TOU map types define attributes shared by TOU maps of a given type.

Prerequisites: You must define TOU map templates and time zones before you can create TOU map types. Refer to the Oracle Utilities Application Framework online help for more information about creating time zones.

To maintain existing TOU map types, select **Admin** > **Consumption** > **TOU Map Type** > **Search** then use standard actions to edit, duplicate, or delete a TOU map type.

To define a new TOU map type, follow these steps:

1. Select **Admin** > **Consumption** > **TOU Map Type** > **Add**

    ➤ **NOTE:** If your system supports more than one TOU map type business object, you will be prompted to elect a business object for this TOU map type.

2. Enter a name and a meaningful description for the TOU map type.
3. Select the business object to use when creating TOU maps of this type
4. Select the time zone for the TOU map type.
5. Specify the interval size for TOU map data created from the map type. Interval size is designated as hours:minutes:seconds (HH:MM:SS)
6. Select the default TOU map template for the TOU map type
7. To add or remove override TOU map templates for this TOU type, click the + or - sign in the Override TOU Map Templates section and select the TOU map template.
8. Click **Save**.

Now you can use the TOU map type when creating TOU maps.

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 .

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 from the **Admin** > **Usage** > **Usage Subscription Type** > **Search**.

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 Service Point Types
- Valid Usage Recipients
- Valid Usage Groups
- Valid Dynamic Option Types

Where Used

Follow this link to open the data dictionary where you can view the tables that reference D1_US_TYPE.

Integrating Usage Transactions

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.

Processing and Sending Usage Transactions to External Systems

When Oracle Utilities Meter Data Management is deployed with Oracle Utilities Customer Care and Billing, usage transaction information, including bill determinants and other data can be sent to 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
- **Override Processing Method**: outbound message types for specific usage subscription types if applicable

For a deeper understanding of integrating usage transactions with 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 deployed with Oracle Utilities Customer Care and Billing that requests bill determinants, usage transaction processing should be coordinated with Oracle Utilities Customer Care and Billing's billing process.

Requests from Oracle Utilities Customer to Meter that have been indicated as "batch requests" 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.

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.

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 batch control (D1-CSPSR) and business object (D1-SPMsrmtCycScheduleRoute).

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 Rule.

For a deeper functional understanding of Usage Calculation, refer to the About Usage Calculation section.

Configuring Usage Transaction Exception Types

This portal is used to display and maintain a Usage Transaction Exception Type.

Refer to Understanding Usage Transaction Exception Types for more information.

You can access the portal from the **Admin** > **Usage** > **Usage Transaction Exception Type** > **Search**.

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

Where Used

Follow this link to open the data dictionary where you can view the tables that reference D1_USAGE_EXCP_TYPE.

Usage Groups

Understanding Usage Groups

Usage Groups are collections of Usage Rules that are used to either calculate bill determinants or validate bill determinants. During the Usage Transaction process, the system executes the Usage Rules defined in the Usage Group referenced on the Usage Subscription. The rules within a Usage Group are defined in a specific sequence, allowing control over the order in which the rules are executed.

Usage Groups are associated with specific Usage Subscriptions and Usage Subscriptions types (or both). When assigned to Usage Subscriptions, Usage Groups contain the Usage Rules to be used to calculate usage / bill determinants. Usage Groups associated with Usage Subscription types are those groups considered valid for Usage Subscriptions of that type.

Usage Groups can also specify a list of Device Configuration types that are considered valid. Usage Groups should only be associated with Usage Subscriptions for Service Points related to Device Configurations of a valid Device Configuration type.

Usage Groups can also be referenced by the Execute Usage Group Usage Rule.

For a deeper functional understanding of Usage Calculation, refer to the About Usage Calculation section.

Configuring Usage Groups

This portal is used to display and maintain a Usage Group.

Refer to Understanding Usage Groups for more information.

You can access the portal from the **Admin** > **Usage** > **Usage Group** > **Search.** 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 Group**: Defines basic information about Usage Group

- **Usage Rules List**: lists the Usage Rules belonging to the group
- **Referencing Usage Rules List**: lists the Usage 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

Where Used

Follow this link to open the data dictionary where you can view the tables that reference D1_USG_GRP.

Usage Rules

Understanding Usage Rules

Usage 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 Rules are created for a specific Usage Group. For example, if you were configuring two Usage Groups and both included a specific Usage Rule, you would need to create two instances of the Usage 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

On almost every Usage 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.

For a deeper functional understanding of Usage Calculation, refer to the About Usage Calculation section.

Configuring Usage Rules

This portal is used to display and maintain a Usage Rule.

Refer to Understanding Usage Rules, Understanding Usage Groups, and Understanding Usage Transaction Exception Types for more information.

You can access the portal from the **Admin** > **Usage** > **Usage Rule** > **Search**. 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 Rule**: Defines the Usage Rule, including parameters used when executing the rule
- **Eligibility Criteria List**: Lists the eligibility criteria defined for the rule

Where Used

Follow this link to open the data dictionary where you can view the tables that reference D1_USG_RULE.

Pre-Calculation Usage Rules

The following is a list of the pre-calculation Usage 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 Rule Name	Applicable Data Type(s)	Purpose
Alignment and Delay Usage Rule	Interval or Scalar	This rule can be used to handle two main needs: 1) aligns all measuring components for a Usage Subscription to the same date (whether on the same device or separate devices) 2) delays the usage transaction until the end of the retry window based on
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 Rules

The following is a list of the calculation Usage 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 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). This rule provides aggregated usage for all selected interval measuring components (filter by TOU, SQI & UOM) associated to a usage subscription. This rule can also multiply total usage by a factor using a custom formula.

Daily Scalar Usage 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. This rule can also be used to provide register readings by measuring component. For consumption values, only the beginning and ending readings are exported 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. This rule retrieves the interval data for measuring components associated to a usage subscription by TOU, SQI and UOM. 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.
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. 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. Note: This rule is used for traditional monthly read meters.
Get Subtractive Interval Details	Get Subtractive Interval Details	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. 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. This rule loops through each tier that is configured and calculates the imbalance amount associated to that tier level. 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.
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. Typical uses: 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. Define complex formulas using various interval curves, profile factor values or calculated service quantities (bill determinant values). Support math functions: sin, cos, square root, etc. Store derived curves in memory that can be used in subsequent calculations Please note, this rule is not as efficient as other rules.

Post-Calculation Usage Rules

Below is a list of the post-calculation Usage 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 Rule Name	Applicable Data Type(s)	Purpose

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 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 Rules

There are Usage 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 Rule Name	Purpose
Execute Usage Group	This rule performs a call to execute a separate Usage Group which includes execution of all Usage 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 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 / 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 Rules

Alignment and Delay Usage Rule

Overview

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

Overview

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.

➤

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 Rules

Apply Math (Interval Data)

Overview

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.

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

Daily Scalar Usage Rule

Overview

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 Rule's UOM/TOU/SQI will be processed.

If configuring this rule with interval and scalar Usage Rules under a Usage Group, it should be executed after interval Usage Rule and before the scalar Usage Rule.

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

Overview

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 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 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 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

Overview

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.

> **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

Overview

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 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.

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 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 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

Overview

This rule is used to calculate the usage for a subtractive interval measuring components. This differs from the Get Interval Data usage 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 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

Overview

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 Rule instance are processed.

Measurements within the period are mapped to time of use quantities based on the TOU map defined on the Usage 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

Overview

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.

➤ **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

Overview

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 his/her 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

Round and Adjust Usage

Overview

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 Service Quantity identifiers, rounding method, and Service Quantity bucket to hold the adjustment. The Usage 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

Overview

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

Post-Calculation Usage Rules

Usage High/Low Rule

Overview

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

Overview

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 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

Overview

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 Rules

Execute Usage Group

Overview

This rule performs a call to execute a separate Usage Group which includes execution of all Usage 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 Group. Creating a separate CALC_KWH_SUM Usage Group for this one calculation allows you to isolate the configuration points. Then the CALC_KWH_SUM Usage Group can be referenced in every other Usage 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

D1-UsageRuleReferredUsageGroup

Exception Handler

Overview

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 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 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 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	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 Rule. However, instead of calculating a single demand value for the entire period it will calculate a demand value for each TOU period:

Usage 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.

Dynamic Option Types

Understanding Dynamic Option Types

Dynamic option types store information common to dynamic options of a specific type.

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 from the **Admin > Usage > Dynamic Option Type**.**Admin** > **Usage** > **Dynamic Option Type** > **Search.**

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

Where Used

Follow this link to open the data dictionary where you can view the tables that reference D1_DYN_OPT_TYPE.

Bill Cycle

Understanding Bill Cycles

Bill cycles identify when a customer is going to be billed and when bill determinants are going to be calculated.

NOTE: Meter Data Management bill cycles are created from Customer Care and Billing bill cycles via Customer To Meter Master Data Synchronization.

Configuring Bill Cycles

This portal is used to display and maintain a Bill Cycle.

Refer to Understanding Bill Cycles for more information.

You can access the portal from other objects such as Measurement Cycles.

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.

NOTE: Meter Data Management bill cycles are created from Customer Care and Billing bill cycles via Customer To Meter Master Data Synchronization. Bill cycles should only be updated via the Customer Care and Billing Bill Cycle portal.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference D1_BILL_CYC.

Defining Time of Use Options

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. For example,

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 from the **Admin** > **Usage** > **TOU Group** > **Search.**

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.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference D1_TOU_GRP.

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.

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 TOUs

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 algorithm type Derive Time Of Use For Date Time (D2-DERTOU-DT) as an example.

Configuring Time of Use Map Templates

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 from the **Admin** > **Usage** > **TOU Map Template** > **Search.**

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.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference D1_TOU_MAP_TMPLT.

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 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 algorithm type Create TOU Map Data (D2-CRETMD-CT) 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 from the **Admin** > **Usage** > **TOU Map Type** > **Search.**

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

- **TOU Map Type List:** This zone lists all TOU Group 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.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference D1_TOU_MAP_TYPE.

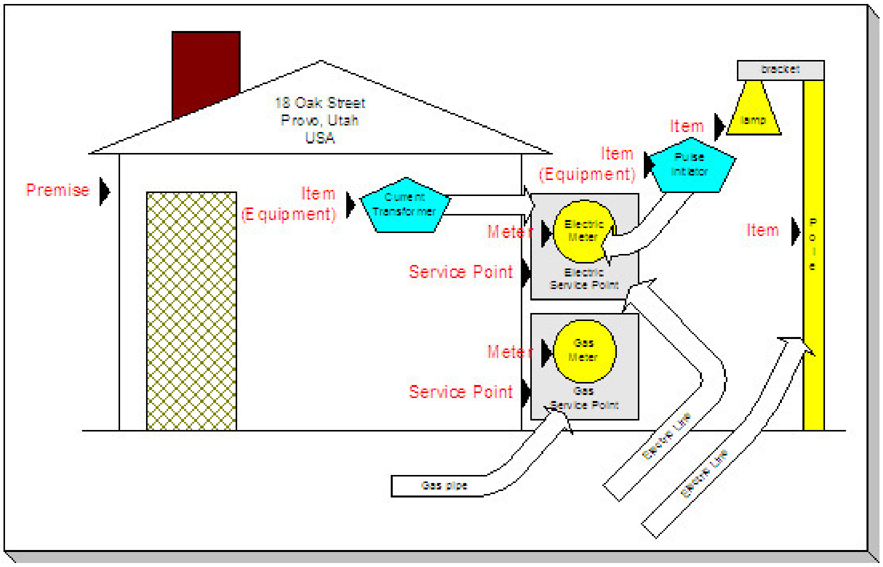
Defining Premise and Service Point Options

A premise is where a customer consumes the services supplied by your company. Located at a premise are the various devices that consume energy and measure consumption. Before you can define premises and devices, you must set up the control tables defined in this section.

WARNING:
The topics in this section do not describe every table that must exist in order to set up premises and service points. Many premise and service point control tables are described in Defining Device Options, and Defining Service Order Options. The tables described in this section are those that must be set up regardless of the type of service.

An Illustration Of A Premise

The following picture illustrates a premise with 2 service points, 2 meters, and 2 badged items:



The following concepts are illustrated above:

Premise A premise describes a location at which your company supplies some type of service. In addition to the obvious address information, a premise also contains geographic coordinates, meter read instructions, and taxation jurisdiction information.

- **FASTPATH:**
  For more information about the control tables that must be set up before you can define a premise see Setting Up Premise Options.

Service Point A service point (SP) is a geographic location at which service(s) are delivered to a premise. The SP record maintains information about the type of service, the measurement cycle (if the service is metered), the field office responsible for maintaining the service, the distribution company that supplies the service, etc.

There are three major categories of service points:

- Those where the rate of consumption and the total amount of consumption is measured (e.g., electricity, gas, water) by a meter. You can think of this type of service point as a "socket" into which a meter can be plugged. Over time, many meters may be plugged into the socket. We refer to these types of service points as metered.
- Those that hold badged items. A badged item is a physical device with a unique identity (e.g., a specific street light, a specific hydrant). You can think of this type of service point as a "socket" into which a badged item can be plugged. Over time, many items may be plugged into the socket. We refer to these types of service points as item-based.
- Those used to hold one or more non-badged items. For example, if your organization doesn't badge street lamps, you can use a single service point to hold an infinite number of lamps. We refer to these types of service points as non-badged.

> ● **FASTPATH:**
> Refer to Service Points (SPs) for more information about non-badged items.

An unlimited number of SPs may exist at a premise. However in reality, the number of SP's is related to the number of services supplied by your company. For example, an electric and gas company will typically have two SPs per premise.

> ● **FASTPATH:**
> For more information about the control tables that must be set up before you can define service points refer to Setting Up Generic Service Point Options, Defining Device Options.

Field activities may be dispatched to all types of service points.

Meter A meter is a physical device used to measure the amount of gas, water, or electricity used by a customer. While most meters measure consumption in a single unit of measure (e.g., gallons, cubic feet, kilowatt-hours), some electric meters are extremely sophisticated and measure several different factors. For example, some electric meters measure how much was used, when it was used, the efficiency of consumption, the maximum amount used, and a few other unusual things.

> ● **FASTPATH:**
> For more information about the control tables that must be set up before you can define a meter, refer to Defining Device Options, and Setting Up Consumption Estimation Parameters.

Item An item that is not consider to be "equipment" is a physical device that does NOT measure consumption, but impacts billing in some way (i.e., there are charges in your rates based on the number and type of items installed at a service point). Examples include street lights, light poles, and security cameras. Items are related to service points and a service point can have one or more items linked to it.

> ● **FASTPATH:**
> For more information about the control tables that must be set up before you can define an item, refer to Configuring Device Types.

> For more information about premises and service points, refer to Understanding The "V".

Setting Up Premise Options

This section describes tables that must be set up before you can define premises.

Defining Premise Types

Open **Admin** > **Geographic** > **Premise Type** to define the premise types used to categorize your premises.

Description of Page

Enter a unique **Premise Type** and a **Description** for every premise type.

Allow CIS Division controls whether premises of this type can use the CIS Division on the premise page. Valid values are Not Allowed, Optional, and Required.

If your implementation requires restricting data by CIS Division, it is recommended that Allow CIS Division be set to Required.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference CI_PREM_TYPE.

Implementing Address Validation

In order to set up address validation, you must ensure that the following Option Types are defined with their corresponding values on the General System Configuration – Feature Configuration page:

Option Type	Description
Address Validation Script	Indicates the name of the BPA script to invoke when a user clicks on the validate address button. This option is required if the Allow Address Validation option is set to Y. The base product includes sample script C1-ValAddr that uses Oracle Spatial procedures and functions to geocode an address and return its matching quality code.
Allow Address Validation	Indicates whether an integration with address validation software is implemented. If set to Y, the system will render a validate button along with the address fields displayed throughout the application to trigger address validation. Valid values are 'Y' and 'N'. If set to Y, the Address Validation Script to invoke must be specified.

As described in the above table, the base product includes sample script C1-ValAddr uses Oracle Spatial procedures and functions to geocode an address and return a matching quality code. To use this script, ensure that the Oracle Spatial Geocoding (F1-ORAGEOCD) algorithm is plugged in on installation options for the geocoding service system event. A data source containing the geocoding data must also exist on the database.

If your implementation uses something other than Oracle Spatial for address validation, you'll need to create a new address validation BPA script to specify on the General System Configuration feature configuration. For additional information about creating this script, see the steps descriptions for the C1-ValAddr script in the application. The new script should contain the following move step to update the page:

- If the calling page is an old style page, the move from the BPA to the page can be achieved by specifying a move step from a source field to a User Interface Field.

- If the calling page is a display map, the move from the BPA to the page can be achieved by specifying a step to read the Business Object, modify the values as returned by the validation software, and then update the BO.

Setting Up Generic Service Point Options

This section describes tables that must be set up before you can define service points.

Setting Up Service Point Types

Every service point must reference a service point (SP) type. The SP type controls almost all aspects of the service point's behavior (e.g., the type of field activity that may be dispatched to it, the type of service agreement that may be linked to it, the type of device that may be installed at it).

The topics in this section describe the windows used to set up your SP types.

> **WARNING:**
> Setting up SP types requires careful analysis of your company's SA types, field activities, and its consumption estimation philosophy. Refer to Designing SP Types for more information about this design process.

Setting Up Premise & Service Point Postal Defaults

You set up postal defaults if your company is able to default field values onto new premises and service points based on the premise's postal code. The topics in this section describe how to maintain postal defaults.

> ● **FASTPATH:**
> For more information about where these default values are used, refer to Maintaining Premises and Maintaining Service Points.

Postal Defaults - Main

To define premise postal defaults, open **Admin** > **Geographic** > **Postal Code Default** > **Add.**

Description of Page

Enter the **Country Code** and range of postal codes to which the default values apply using the **From Postal Code** and **To Postal Code**.

> ➤ **NOTE:**
> You may not have postal defaults whose from / to postal codes overlap.

Trend Area is not applicable for Customer to Meter.

Enter the **County** to be defaulted onto new premises located in this postal code range.

Enter the **City** to be defaulted onto new premises located in this postal code range.

Enter the **CIS Division** to be defaulted onto new premises located in this postal code range.

Enter the **State** to be defaulted onto new premises located in this postal code range.

Enter the **Time Zone** to be defaulted onto new premises located in this postal code range.

Use the **Characteristic Types and Values** collection to define the **Characteristic Type**s and their respective **CharacteristicValue**s to be defaulted on premises located in this postal code range. In addition to providing interesting information, these characteristics may also determine the prices and tax rates on the bills generated for the services consumed at a premise.

- **FASTPATH:**
  For more information about characteristics, see Setting Up Characteristic Types & Their Values and An Illustration Of A Bill Factor And Its Characteristics.

Use the **Geographic Types and Values** collection to define the **Geographic Type**s and their respective **Value**s to be defaulted on premises located in this postal code range. In addition to providing interesting information, these values may be used to sort field activities in geographic value order.

Where Used

This information is defaulted when a new Premise is added. Characteristics and geographic values are also defaulted when the postal code for a Premise is changed. Refer to Maintaining Premises for more information.

Postal Defaults - Service Default

To define values to be defaulted for a service point located in a postal range, open **Admin** > **Geographic** > **Postal Code Default** > **Search** and navigate to the **Service Defaults** page.

- **FASTPATH:**
  For more information about where these default values are used, refer to Maintaining Service Points.

Description of Page

Use the **Service Defaults** collection to define values to be defaulted on service points located in a given postal code range belonging to a given **Service Type** (note, a service point's Service Type comes from its SP Type).

Stock Location is not applicable for Customer to Billing.

- Use the **Characteristic Types and Values** collection to define the **Characteristic Type**s and their respective **CharacteristicValue**s to be defaulted on new service points located in this postal code range. In addition to providing interesting information, these characteristics may also determine the prices and tax rates on the bills generated for the services consumed at a service point.
- Use the **Geographic Types and Values** collection to define the **Geographic Type**s and their respective **Value**s to be defaulted on new service points located in this postal code range. In addition to providing interesting information, these values may be used to sort field activities in geographic value order.

Designing SP Types

Every service point must reference an SP type. When you set up an SP type, you define how the system manages many aspects of its service points' behavior.

> **NOTE:**
> **Perfect foresight.** In a perfect world, the other control tables would have been set up with perfect foresight of setting up your SP types. In reality, setting up your SP types may invalidate some of your earlier decisions. Don't feel bad if this happens, some amount of iteration is natural.

Designing your SP types is an iterative process. To minimize the number of iterations, we recommend using the steps outlined in this section to complete the following table. When the table is complete, you're ready to set up your SP types.

SP Type	Service Type

Service Segmentation

At a minimum, you will have one SP type for every different type of service that exists at your premises. If we assume that your organization sells gas, water, waste water and electricity services, you will need four SP types.

SP Type	Service Type
GAS	Gas service
WATER	Water service
WASTE WATER	Waste water service
ELECTRIC	Electric service

Device Segmentation

For each service, determine if there exist meters, badged items, or non-badged items at the service points. For example, if we assume:

- Electric service has service points with meters, badged lamps, and parking lots.
- Gas and water services just use meters.
- Waste water service doesn't actually have a meter installed at it (it uses the water consumption measured by the water service point's meter)

Then your SP types will be:

SP Type	Service Type	SP Sub Type

GAS - METERED	Gas service	Meter
WATER - METERED	Water service	Meter
WASTE WATER	Waste water service	Unbadged
ELECTRIC - METERED	Electric service	Meter
ELECTRIC - BADGED LAMP	Electric service	Item
ELECTRIC - PARKING LOT	Electric service	Unbadged

SA Type Segmentation

Every SP Type whose service points can be linked to a service agreement has one or more SA types. These define which type of service agreements can pay for the service point's service. If different service points have different valid SA types, you will need to split the SP types accordingly.

The SA Type segmentation of SP Types is the most complicated design decision you'll have to make. Unfortunately, the decision making process is subjective and iterative. The iterations are caused by the fact that the number of SA types is dependent on the number of SP types (and vice versa). We recommend the following to help work your way through this conundrum:

- Design your SA types using the information in Defining Service Agreement Types.
- Return to your SP types and determine if, given the proposed SA types, you can define a list of valid SA types for each SP type. If you find the population of SA types (and their valid rates) could result in invalid rates paying for service at a service point, divide your SA types further.

Defining Bill and Measurement Cycles

This chapter is dedicated to issues related to defining cycles, routes and schedules in the system.

Defining Bill and Measurement Cycles

Every account references a bill cycle. An account's bill cycle controls when it is billed.

Every manually read metered service point references a measurement cycle. A service point's measurement cycle controls when a service point's meter is measured.

The design of your measurement and bill cycles is inextricably linked because you probably want to bill your customers shortly after their meter devices are measured.

In this section, we describe how to design and set up these cycles. In addition, we discuss how to set up bill period schedules. These are used to define the bill segment end date for special types of non-metered service agreements.

> **NOTE:**
> **Recommendation.** We recommend reviewing Bill Frequency - Bill Cycle vs Bill Segment Duration before setting up this information.

The Big Picture of Bill Cycles, Measurement Cycles, and Bill Periods

The topics in this section provide background information about a variety of bill cycle, measurement cycle, and bill period issues.

Designing Cycles for Metered Services

The topics in this section provide background information about a variety of measurement cycle issues.

A Description of the Cyclical Meter Read Process

Manual meter readers using handheld devices record most meter reading measurements in the field. These meter reads are uploaded into the system for use by Oracle Utilities Meter Data Management.

A service point's installed meter device is measured due to the following data relationships:

- Every metered service point references a schedule route.

- A route references a measurement cycle.
- A measurement cycle has measurement cycle schedules that define when the service points in the cycle are measured.
- Every measurement cycle schedule contains two dates:
  - **Scheduled selection date.** This is the date Meter Data Management could use to select the service points in the cycle for download to your handheld software via custom algorithms to create handheld activities when pending measurement cycle schedules are proposed (refer to Measurement Cycle and Bill Determinants for more information).
  - **Expected work date.** This is the date the meter is expected to be read.

➤ **NOTE:**

Overriding a route within a specific schedule. Rather than downloading all routes within a cycle, you can set up the system so that only specific routes are downloaded on any given date. This is a very powerful feature. You can use it, for example, to estimate specific cycles every other month or indicate the customer reads the meter every third month. Refer to Designing Measurement Cycles, Routes, And Schedules for more information.

- On the scheduled selection date, Meter Data Management could create a download file, via custom algorithms, containing information to be sent to your handheld software. This download file should contain information about every register or measuring component on every meter in the routes being downloaded.
- Your handheld software distributes this information to the handheld devices and then your meter readers do their job. When they finish, the resultant measurements are uploaded into the system for subsequent use by other processes such as usage transactions for billing.

WARNING:

It is very important to create a measurement cycle schedule for every expected read date regardless of whether the cycle's routes are downloaded. Why do you have to do this? Because Meter Data Management uses the expected work date on the measurement cycle to know when to look for a measurement during usage calculations. If it can't find a reading on or near this date, Oracle Utilities Meter Data Management may estimate consumption. Without a measurement cycle schedule, MDM wouldn't know when to look for readings. So, for example, if you estimate a given cycle's consumption EVERY OTHER MONTH, you must create a measurement cycle schedule for EVERY month. On each month's schedule, you must define if the routes should be downloaded.

● **FASTPATH:**

For more information about the how to control when a cycle is read, refer to Setting Up Measurement Cycle Schedules.

Designing Measurement Cycles, Routes, And Schedules

The topics in this section provide information describing how to design your measurement cycles, routes, and schedules.

- Designing Measurement Cycles For Meter Reading
- Designing Schedule Routes For Meter Reading
- Designing Metered Measurement Cycle Schedules

Designing Measurement Cycles for Meter Reading

The criterion that affects the number of measurement cycles has nothing to do with when meter readers physically read your meter devices. Rather, the frequency that you bill the meter's consumption (real or estimated) is what controls the number of measurement cycles.

So, for example, if you bill every month, but read every OTHER month, you'll have 20 measurement cycles - one for each bill day during the month. If you bill bimonthly, you'll have 40 measurement cycles. If you bill quarterly, you'll have 60 measurement cycles. Etc.

➤

NOTE:
Different billing frequencies are possible for different service points. If you have different billing frequencies for your different types of metered service, you'll need a different set of cycles for each billing frequency. For example, if you bill water quarterly and electricity monthly, you'll have 20 bill cycles (one for each bill day during a month), but you'd need 60 quarterly measurement cycles for your water service points, and 20 monthly measurement cycles for your electric service points. This would result in a customer getting billed every month. However, four times a year, their bill would contain a charge for both electricity and water.

Top of the Page

Designing Service Routes For Meter Reading

The following points describe the relationship between a meter read, a route and a measurement cycle:

- A measurement cycle contains one or more routes.
- A route has one or more service points.
- A service point holds an installed meter device.
- And a device is what is measured.
- Therefore, the number of devices a person can read in a day limits the number of service points in a route.

WARNING:
If your company supplies electric service and uses MV90's, you will need to take advantage of "notional" routes. A "notional" route's service points are never actually read by a human. Rather, the service points' consumption is fed to the system by a sophisticated device (e.g., an MV90). These service points still need to be linked to a route because usage calculation is dependent on the route's cycle to determine the expected meter read date.

Top of the Page

Designing Metered Measurement Cycle Schedules

The process of designing your measurement cycle schedules is either easy or complicated. It will be easy if all routes within a cycle are downloaded when the measurement cycle is scheduled for download. It will be complicated if you download a subset of routes within a cycle on any given download date. We'll provide a few examples to help explain why.

If you download all cycles in a route whenever the cycle is downloaded, your measurement cycle schedule will look as follows.

➤

NOTE:
Bill cycles. We've included each measurement cycle's related bill cycle to help you understand when the measurement cycle's consumption will be billed. Bill cycles are discussed in Bill Cycles.

Measurement Cycle	Download Date	Expected Work Date	Which Routes To Download	Bill Cycle	Bill Window	Estimation Date
1	28-May-99	31-May-99	All	1	31-May-99 to 2-Jun-99	2-Jun-99
2	31-May-99	1-Jun-99	All	2	1-Jun-99 to 3-Jun-99	3-Jun-99
3	1-Jun-99	2-Jun-99	All	3	2-Jun-99 to 4-Jun-99	4-Jun-99
4	2-Jun-99	3-Jun-99	All	4	3-Jun-99 to 7-Jun-99	7-Jun-99
Etc. to 20						

Now let's complicate life. If we assume you physically read your routes every other month (but bill monthly using estimated consumption), then you'll need the following measurement cycle schedule.

Measurement Cycle	Download Date	Expected Work Date	Which Routes	Bill Cycle	Bill Window	Estimation Date
1	30-May-99	31-May-99	1, 2, 3 - Download 4, 5, 6 - Estimate	1	31-May-99 to 2-Jun-99	2-Jun-99
2	31-May-99	1-Jun-99	1, 2, 3 - Download 4, 5, 6 - Estimate	2	1-Jun-99 to 3-Jun-99	3-Jun-99
3	1-Jun-99	2-Jun-99	1, 2, 3 - Download 4, 5, 6 - Estimate	3	2-Jun-99 to 4-Jun-99	4-Jun-99
4	2-Jun-99	3-Jun-99	1, 2, 3 - Download 4, 5, 6 - Estimate	4	3-Jun-99 to 7-Jun-99	7-Jun-99
Etc. to 20						
1	29-Jun-99	30-Jun-99	1, 2, 3 - Estimate 4, 5, 6 - Download	1	30-Jun-99 to 2-Jul-99	2-July-99
2	30-Jun-99	1-July-99	1, 2, 3 - Estimate 4, 5, 6 - Download	2	1-July-99 to 3-Jul-99	3-July-99
3	1-July-99	2-July-99	1, 2, 3 - Estimate 4, 5, 6 - Download	3	2-July-99 to 4-July-99	4-July-99

4	2-July-99	3-July-99	1, 2, 3 - Estimate 4, 5, 6 - Download		4	3-July-99 to 7-July-99	7-July-99
Etc. to 20							

Notice the following:

- You still have 20 measurement cycles even though you only read the meters every other month. Why? Because Oracle Utilities Meter Data Management uses the expected work date on the measurement cycle to know when to look for a measurement. If it can't find a reading on this date, Meter Data Management estimates consumption. Without a measurement cycle schedule, Meter Data Management wouldn't know when to look for readings.
- Every other month you download half the routes in each cycle and estimate consumption for the other half.
- If you don't download all routes when a measurement cycle is scheduled, you have to indicate how to handle every route in the cycle.

The above design is infinitely flexible. You can use it to handle any number of requirements:

- Estimate consumption every seventh month.
- Bill every month, but only read once a quarter.
- Etc.

Bill Cycles

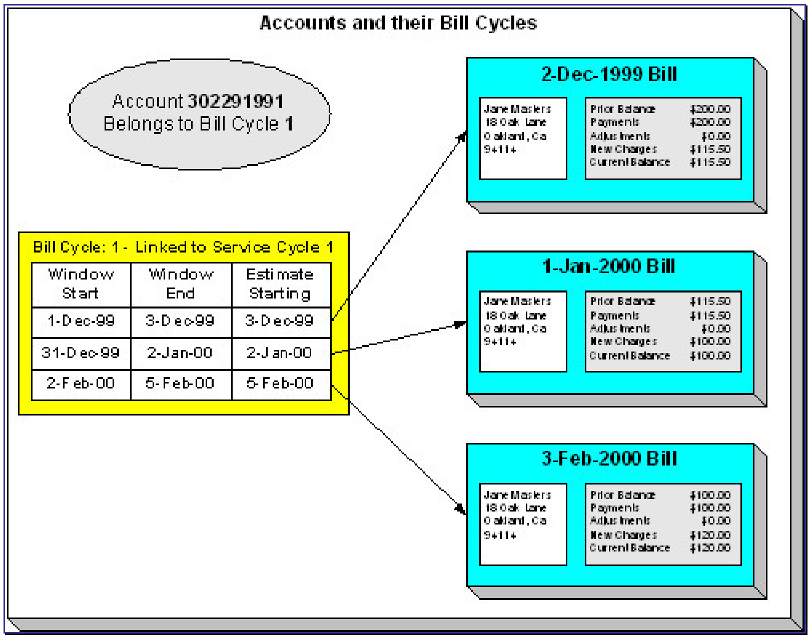
The topics in this section provide background information about a variety of bill cycle features.

The Cyclical Billing Process & Window Billing

The cyclical bill creation process creates most bills. This process works as follows:

- Every account references a bill cycle. The bill cycle's schedule controls WHEN the system attempts to create bills for the account.
- Every bill cycle has a bill cycle schedule that defines the dates when a cycle's accounts are to be billed. Rather than attempt to create bills on one evening, most bill cycles use a concept of "window billing" where the system attempts to produce bills for accounts over a few nights. The first night (i.e., the day the window opens) should be the earliest day on which meter reads for the account can enter the system. The last night (i.e., the day the window closes) should be the last possible day in which a meter read can enter the system. This concept of window billing allows you to start billing accounts on the earliest possible day and then bill the stragglers over successive evenings. This results in much better cash flow.
- When the bill cycle creation process runs, it looks for bill cycles with open bill windows. It then attempts to create bills for the accounts in each such cycle. If a bill is created, it will send it out that evening. If a bill cannot be created, the system will create a bill in the "error" state with a message that can be analyzed by your billing staff. The next day, your staff can either correct the problem or not. When the bill cycle creation process next runs, it deletes all "error" bills and attempts to recreate them. It continues this throughout the bill window. If bills are in error at the end of the window, they will remain in this state until a user fixes them. If the bills are still in error when the cycle's next window opens, a billing error will be generated.

The following diagram should help clarify the above.



●

FASTPATH:
For more information about the how to control when bills are produced for a cycle, refer to Bill Cycle - Bill Cycle Schedule.

Designing Your Bill Cycle

The number of bill cycles is determined by the frequency that you bill your customers. So, for example, if you bill every month, you'll have 20 bill cycles - one for each bill day during the month. If you bill bimonthly, you'll have 40 bill cycles. If you bill quarterly, you'll have 60 bill cycles. Etc.

Keep in mind the following:

- You may need additional bill cycles if you allow customers to be billed off-cycle. For example, you could create a bill cycle called "Seniors" and link this to every senior citizen. You would set up this bill cycle's schedule to create bills shortly after social security checks are issued.
- You may need other bill cycles for customers with non-metered services (e.g., land leases, one time invoices).

△

IMPORTANT:
An account's bill cycle should attempt to create bill segments at least as often as the shortest service agreement duration. For example, if an account has both monthly and quarterly service agreements, the account should be placed on a monthly bill cycle. Refer to Bill Frequency - Bill Cycle vs Bill Segment Duration for more information.

The Relationship Between Metered Measurement Cycles and Bill Cycles

As you can deduce, a service point's measurement cycle is related to an account's bill cycle.

The following table is an example of how you would set up the various dates on the various schedules that control meter reading and billing.

Measurement Cycle	Download Date	Expected Work Date	Bill Cycle	Bill Window	Estimation Date
1	28-May-99	31-May-99	1	31-May-99 to 2-Jun-99	2-Jun-99
2	31-May-99	1-Jun-99	2	1-Jun-99 to 3-Jun-99	3-Jun-99
3	1-Jun-99	2-Jun-99	3	2-Jun-99 to 4-Jun-99	4-Jun-99
4	2-Jun-99	3-Jun-99	4	3-Jun-99 to 7-Jun-99	7-Jun-99
Etc.					

Notice the following about this example:

- The bill cycle code is the same as the measurement cycle. This is not necessary, it's just good practice.
- The bill window starts on the first date on which a meter read could be uploaded.
- The bill window ends on the day after the last possible day a read could be uploaded for the bill cycle.
- Oracle Utilities Meter Data Management only estimates consumption on the last day of the bill window.

FASTPATH:
For more information about the bill cycle schedule, refer to Setting Up Bill Cycles.

How Does An Account Get Its Bill Cycle?

Most accounts are created behind-the-scenes when a user uses the "add account" option on Person - Main Information. An account created like this doesn't have a bill cycle. Rather, it sits in limbo awaiting the activation of its first service agreement. When a service agreement is activated, the system populates the account's bill cycle using the following algorithm:

- If an account's bill cycle is protected, the activation of a service agreement will not affect an account's bill cycle. Refer to Protecting An Account's Bill Cycle for more information.
- If the service agreement being activated is for metered service, the account is given a bill cycle that will generate bills shortly after the service point is read. The route the system follows to get this bill cycle is a bit indirect:
  - A metered service agreement references one or more metered service points.
  - Every metered service point references a measurement cycle (the measurement cycle controls when the meter at the service point is read).
  - Every measurement cycle references a default bill cycle. It is this bill cycle that is populated on the account paying for service at the service point.
- If the service agreement being activated is for metered service and the associated service point is externally managed along with its measurement cycle and bill cycle, an external call is made to obtain the bill cycle from Meter Data Management. This bill cycle is assigned to the service agreement’s account.

- If the service agreement being activated is not metered, the system cannot populate the account's bill cycle because there is no measurement cycle from which to default the bill cycle. This means the account's bill cycle will be blank until a user specifies a bill cycle for the account (using the Account page).

➤ **NOTE:**
A To Do entry highlights accounts without a bill cycle. A To Do entry highlights those accounts that require a user to specify a bill cycle. This entry is generated for accounts without a bill cycle that have active service agreements.

When the last service agreement linked to an account is stopped, the account's bill cycle will be changed so that the account will be final billed when billing next executes. Refer to What Happens At Finalization Time? for more information.

When a service point's measurement cycle is updated, and the account's bill cycle is not protected, the system automatically updates the account's bill cycle to be in sync with the measurement cycle. Note that this will only take place if the Allow Bill Cycle Synchronization **Option Type** on the General System Options Feature Configuration is set to Y.

When an externally managed service point's measurement cycle is updated that results to a change in the bill cycle, a change notification is sent by Meter Data Management to trigger an corresponding update of the associated account’s bill cycle in Customer Care and Billing. Note that this will only take place if the Allow Bill Cycle Synchronization **Option Type** on the General System Options Feature Configuration is set to Y.

Protecting An Account's Bill Cycle

Over time, as a customer moves from premise to premise, their bill cycle is changed behind-the-scenes to be in sync with the latest service point's measurement cycle (as described in the previous section). If you do not want the system to change an account's bill cycle when a metered service agreement is activated, you need to turn on the account's **protect bill cycle** flag. You would do this if a customer liked to be billed at the start of the month regardless of when their device was read.

When the last service agreement for an account is stopped, the protect bill cycle flag is reset. This is to ensure that if the customer returns to start new service again, the bill cycle can be set based on How Does An Account Get Its Bill Cycle.

What Happens If An Account Has Service Agreements Spanning Metered Measurement Cycles?

A single account can have service agreements that are in several measurement cycles. The bill cycle on such an account will default based on the last activated metered service agreement.

It's important to be aware that an account will only have bills created when its bill cycle schedule so indicates. This means that the consumption for some service points could remain unbilled for a few weeks.

Designing Bill Periods

Bill periods are used by non-metered service agreements whose bill end dates need to fall on strict dates. You need only set up this information if you have this type of service agreement.

● **FASTPATH:**

> Refer to Ways To Control The End Date Of A Bill Segment for more information.

Every bill period has a calendar that defines when bill segments are created for service agreements that use the bill period. Examples of bill periods include:

- Quarterly Bill - Last Day Of Quarter
- Quarterly Future Bill - Last Day Of Quarter
- Monthly - 15th Day Of Month.
- Monthly Future Bill - Last Day Of Month.

The Quarterly Bill - Last Day Of Quarter bill period would have a schedule that looked as follows:

Earliest Date On Which To Create A Bill Segment	Bill End Date
1-Oct-1998	30-Sep-1998
1-Jan-1999	31-Dec-1998
1-Apr-1999	31-Mar-1999
1-Jul-1999	30-Jun-1999
...	

The Quarterly Future Bill - Last Day Of Quarter bill period would have a schedule that looked as follows:

Earliest Date On Which To Create A Bill Segment	Bill End Date
15-Dec-1998	31-Mar-1999
15-Mar-1999	30-Jun-1999
...	

The remainder of this section provides examples using the above calendars.

The following example assumes the following:

- The service agreement starts on 18-Dec-1998.
- The service agreement's SA type references the Quarterly Future Bill - Last Day Of Quarter bill period.

The following table shows when bill segments will be produced (assuming the account's bill cycle attempt to create segments as soon as possible) for several bill periods:

Earliest Date Segment Will Be Produced	Bill Period
18-Dec-1998	18-Dec-1998 thru 31-Mar-1999
15-Mar-1999	1-Apr-1999 thru 30-Jun-1999
...	

The following example assumes the following:

- The service agreement starts on 18-Dec-1998.
- The service agreement's SA type references the Quarterly Bill - Last Day Of Quarter bill period.

The following table shows when bill segments will be produced (assuming the account's bill cycle attempts to create segments as soon as possible) for several bill periods:

Earliest Date Segment Will Be Produced	Bill Period
1-Jan-1999	18-Dec-1998 thru 31-Dec-1998
1-Apr-1999	1-Jan-1999 thru 31-Mar-1999
...	

FASTPATH:
Refer to Setting Up Bill Periods for information about how to define this information.

Setting Up Bill Cycles

An account references a bill cycle. The bill cycle defines when the account is billed and when the account's debt is checked to determine if it's overdue. To define a bill cycle and its bill cycle schedule, open **Admin** > **Billing** > **Bill Cycle** > **Add.**

Description of Page

Enter a unique **Bill Cycle** and **Description** for every bill cycle.

If your implementation requires restricting bill cycles to accounts by CIS Division, specify a **CIS Division** on the bill cycle.

Use the **Bill Cycle Schedule** collection to define when bills are produced for the accounts in a given bill cycle. The following fields are required for each instance:

Window Start Date Specify the date on which the system should start trying to create bills for accounts in the cycle.

Window End Date Specify the last day on which the system will create bills for accounts in the cycle. This should be the last possible date on which a meter read could be loaded into Oracle Utilities Meter Data Management for the account.

Accounting Date Specify the financial date associated with the bills' financial transaction. The accounting date defines the financial period(s) to which the bills will be booked in your general ledger.

Estimate Date The date on which Oracle Utilities Meter Data Management may start estimating consumption if a real measurement cannot be found. When specified, this date is typically on or shortly before the window end date.

Freeze and Complete Turn on this switch if the system should freeze and complete any bill that is created without errors. If this switch is turned off, all bills created by the billing process will be left in the unfinished state. You would only turn this switch off if you want to verify an entire bill run prior to freezing it (e.g., if you are introducing a new version of a rate). If you turn this off, you will need to return to this page after verifying a bill run and turn it back on for the customers to receive bills. When the system next runs, it deletes all unfrozen bills and recreates them as per the instructions on the bill cycle schedule.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference CI_BILL_CYC.

The batch bill creation process uses this schedule to define the bill cycles for which it should create bills.

NOTE:
Don't forget. After you set up the bill cycles that correspond with measurement cycles, go to Setting Up Measurement Cycle Schedules and update the measurement cycles accordingly.

Setting Up Service Cycles And Measurement Cycle Schedules

Refer to Measurement Cycles, Measurement Cycle and Bill Determinants, and Setting Up Measurement Cycle Schedules sections for more information

Setting Up Measurement Cycle Schedules

- **FASTPATH:**
  Refer to Designing Cycles for Metered Services for more information about measurement cycle. Refer to The Relationship Between Metered Service Cycles and Bill Cycles for more information about how measurement cycles are linked to bill cycles.

Description of Page

When you want to add a new measurement cycle schedule, you must specify the following information:

Service Cycle Enter the measurement cycle ID of the cycle whose routes will be downloaded.

Scheduled Selection Date Specify the date on which the system is meant to download information about the cycle's meters. This date should be a day or two before the scheduled work date.

Define the date the meters in the route are scheduled to be read using **Scheduled Work Date**. This date is extremely important as billing uses it when it looks for meter reads for service points in this cycle. If billing can't find a reading, consumption will be estimated as of this date (assuming the service agreement allows estimation).

Click **Pre-Generate Routes** if you need to finesse the cycle's routes on this Scheduled Selection Date (e.g., because some of the routes shouldn't be downloaded on a given date because the system is meant to estimate consumption). If you don't click this button, the system will create the routes on the scheduled selection date. It does this by creating a route extract for each downloadable route within the cycle (as defined by the route type). If you click this button, the system shows all routes within the cycle in the grid below. You must then define how the system is supposed to download each route on the scheduled selection date.

Click **Delete Routes** if you have pre-generated the routes and you was to remove them and allow the system to create the routes on the scheduled selection date.

Use the **Service Schedule Routes** to define how individual routes within the cycle should be handled during the download. The following fields are required for each schedule read date:

Service Route The unique ID of the service route. The route's route type is displayed adjacent.

Schedule Type This defines if and how the system is supposed to download the route's meter reads. Valid values are: Cust Read, Download, Estimate. Only those routes defined as Download will have meter reads downloaded. The other values are used to document why the route won't be downloaded.

Schedule Status This defines the download status of the route. Valid values are Pending and Complete. This value is only displayed for routes with a schedule type of Download.

Where Used

This information is used by the meter read download process to determine which meter reads to download.

This information is used by the billing process to determine the date on which it expects to find a read. If it cannot find a read on or around this date, and it's OK to estimate consumption, consumption will be estimated as of the scheduled work date. The system uses the service agreement's rate schedule's frequency to determine the period of time around the scheduled work date in which it looks for a read. Refer to Defining Frequency Codes for more information.

Setting Up Bill Periods

Some SA types reference a bill period. The bill period defines when the service agreement's bill segments are produced and the respective end date of each bill segment.

- **FASTPATH:**
  Refer to Designing Bill Periods for more information.

To define a bill period and the bill period schedule, open **Admin** > **Billing** > **Bill Period** > **Add.**

Description of Page

Enter a unique **Bill Period** and a **Description** for every bill period.

Use the **Bill Period Schedules** collection when the system should create bill segments for service agreements that use a given bill period. It also defines the end date of each respective bill segment. The following fields are required:

Bill Date Specify the earliest date on which the system is allowed to create a bill segment for service agreements using this bill period.

Bill Seg End Date Specify the end date of the bill segment. For future bills, this will be after the bill date. For retro bills, this will be before the bill date.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference CI_BILL_PERIOD.

This information is used by the bill segment creation process to determine the end date of service agreements that use a bill period.

Defining Statement Cycles

If you have persons set up in the system to receive statements with financial information, you will need to assign them to a statement cycle and define a schedule for the statement cycle.

- **FASTPATH:**
  Refer to The Big Picture of Complex Statements for more information about statements.

The Big Picture Of Statement Cycles

A statement cycle has a similar purpose to that of a bill cycle. It controls when statements will be produced.

The Cyclical Statement Process

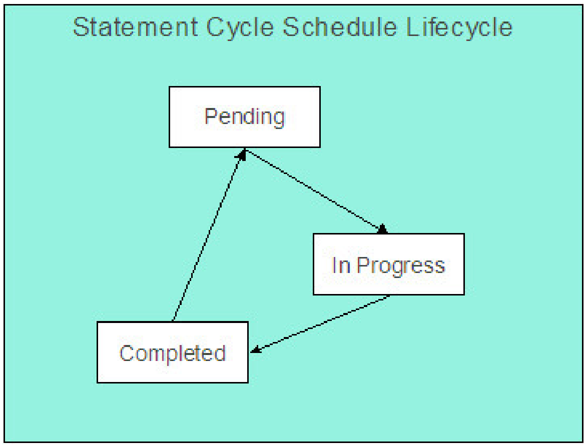
Persons who wish to receive statements will work with you to determine how often these statements should be produced. Some persons may want a monthly statement, some a quarterly and some annually. For each unique schedule that is designed for your various statement persons, you will set up a Statement Cycle and its schedule.

Designing Your Statement Cycles

The number of statement cycles is determined by a combination of the frequency that you will send statements to the statement persons and how many statement cycles you wish to manage within the same frequency.

So, for example, for all the statement persons who wish to receive a monthly statement, will you create only one monthly Statement Cycle so that all monthly statements are produced the same day? Or will you have several monthly statement cycles scheduled throughout the month? The answer will depend on the volume of statements being produced and on how you want to manage the statement production.

Lifecycle of a Statement Cycle Schedule



Pending The statement cycle schedule is added in this state. The Create Statements background process find records in this state to process on the appropriate date.

- **FASTPATH:**
  Refer to Create Statements Background Process for more information.

In Progress Records in this state are currently being processed by the Create Statements background process.

Completed Records in this state have already been processed by the Create Statements background process. If a problem occurs with the Create Statements background process and it needs to be rerun, simply change the status back to pending and rerun the process.

Setting Up Statement Cycles

A Statement Construct references a statement cycle. The statement cycle defines when the statement person will receive statements with financial information related to the accounts and service agreements linked to the statement construct. To define a statement cycle and its statement cycle schedule, open **Admin** > **Billing** > **Statement Cycle** > **Add.**

Description of Page

Enter a unique **Statement Cycle** and **Description** for every statement cycle.

Use the **Statement Cycle Schedule** collection to define when statements are produced for the persons with statement construct records in the given statement cycle. The following fields are required for each instance:

Processing Date Specify the date on which the system should create statements for persons with statement construct records in the cycle.

Status Indicates the status of the cycle schedule. Refer to Lifecycle of a Statement Cycle Schedule for more information.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference CI_STM_CYC.

The batch statement creation process uses this schedule to determine which statement cycles for it should create statements for the statement construct records. Refer to Create Statements Background Process for more information.

Defining Service Agreement Type (SA Types)

Every service agreement must reference a SA type. The SA type defines what you sell, how much you sell it for, to whom you sell it, how overdue debt is collected, and how sales will be booked in your general ledger.

NOTE:
Perfect foresight. In a perfect world, the other control tables would have been set up with perfect foresight of setting up your SA types. In reality, setting up your SA types may invalidate some of your earlier decisions. Don't feel bad if this happens, some amount of iteration is natural.

Designing SA Types For Service Agreements With Service Points

Designing your SA types is an iterative process. To minimize some iterations, we recommend using the steps outlined in this section to complete the following table. When the table is complete, you're ready to set up your SA types.

Division/SA Type	Service Type

The topics in this section provide guidelines describing how to fill in this table for SA types associated with service agreements that charge for service point-oriented services.

CIS Division Segmentation

A CIS division is typically associated with a jurisdiction. The definition of a jurisdiction is a geographic-oriented entity with unique business rules. For example, if you conduct business in California and Nevada, and each state has different collection rules, you will need a separate jurisdiction for each state. You must set up a CIS division for each jurisdiction in which you conduct business.

If we assume that you are located in a single jurisdiction - say California - we will need a single CIS division for all of our SA types.

CIS Division/SA Type
CA

Service Segmentation

At a minimum, you will have one SA type for every different type of utility service offered by your organization. If we assume you sell electricity, gas, water, wastewater and cable; your SA Types will be as follows:

CIS Division/SA Type	Service Type
CA/G	Gas service
CA/W	Water service
CA/E	Electric service
CA/WW	Waste water service
CA/CABLE	Cable

Receivable Segmentation

Many organizations segregate their receivable balances in the general ledger. For example, the receivable amount associated with gas and water service may be maintained in separate GL accounts.

If your organization does this, you will probably have at least one SA type for each such receivable account because each SA type references a distribution code that typically contains the receivable account.

- The word *probably* is underlined because this is a rule of thumb. There are situations where the number of receivable accounts isn't directly related to the number of SA types. This happens when an organization maintains very detailed receivable accounts in the general ledger and maintaining a one-to-one relationship between SA types and distribution codes would lead to a massive proliferation of SA types (and you don't want this!). If your organization maintains very detailed receivable accounts, please speak to your implementers, they should be able to introduce a small customization to generate the appropriate receivable account rather than extract it from the distribution code.
- The word *typically* is underlined because there are several SA types that don't book to a receivable account when bill segments are generated. For example, company usage and charitable contributions. Refer to Company Usage Segmentation and Charitable Contribution Segmentation for examples of SA types that don't book to receivable accounts.

We'll simplify our example and assume your organization has one receivable account for all types of utility service. Given this, we won't need additional SA types to support receivable segmentation:

CIS Division/ SA Type	Service Type	Distribution Code
CA/G	Gas service	A/R-UTIL
CA/W	Water service	A/R-UTIL
CA/E	Electric service	A/R-UTIL
CA/WW	Waste water service	A/R-UTIL
CA/CABLE	Cable service	A/R-UTIL

Revenue Segmentation

Look at your rates and determine which rates can be used by each SA type. The following table shows the sample rates that can be used for each service:

CIS Division/ SA Type	Service Type	Distribution Code	Rates
CA/G	Gas service	A/R-UTIL	GALL-1
CA/W	Water service	A/R-UTIL	WALL-1
CA/E	Electric service	A/R-UTIL	ERES-1, ERES-2, ECOM-1, EIND-1, ELAMP-1
CA/WW	Waste water service	A/R-UTIL	WWALL-1
CA/CABLE	Cable service	A/R-UTIL	CABLE

Now, look at the rates' calculation rule GL Distribution window. You're looking for calculation rules whose GL distribution is affected by revenue class. If there are no revenue classes referenced on the calculation rules, this means that the revenue associated with the rate will be booked to a single GL account regardless of the type of customer. If you see revenue classes, this means that the revenue account associated with the calculation rule(s) differs depending on the SA type's revenue class. If revenue classes are used in the rates, you must create a different SA Type for every revenue class.

Let's assume the following:

- The gas rate (GALL-1) references the RESIDENTIAL, COMMERCIAL and INDUSTRIAL revenue classes in order to differentiate revenue based on the type of customer.
- None of the other rates differentiate revenue based on customer class.

Our SA types will now look as follows:

CIS Division/ SA Type	Service Type	Distribution Code	Revenue Class	Rates
CA/G-RES	Gas service	A/R-UTIL	R - residential	GALL-1
CA/G-COM	Gas service	A/R-UTIL	C - commercial	GALL-1
CA/G-IND	Gas service	A/R-UTIL	I - industrial	GALL-1
CA/W	Water service	A/R-UTIL	N/A	WALL-1
CA/E	Electric service	A/R-UTIL	N/A	ERES-1, ERES-2, ECOM-1, EIND-1, ELAMP-1
CA/WW	Waste water service	A/R-UTIL	N/A	WWALL-1
CA/CABLE	Cable service	A/R-UTIL	N/A	CABLE

Notice that we created new SA types for gas in order to specify the respective revenue class. We didn't do this for the other services because it isn't necessary. However, you should feel free to do this if it feels right or if you need it for reporting purposes. For example, if you want to report on all *residential* service agreements, but you differentiate only gas by residential versus commercial, you'll be in trouble.

Rate Segmentation

Every SA Type whose service agreements have their charges calculated with a rate must have one or more rates linked to it. These define which rates can be linked to the SA type's service agreements. If different service agreements have different valid rate combinations, you will need to split the SA types accordingly.

For example, look at the electric rates. If we assume:

- the RES rates can only be used for residential customer,
- the IND rates can only be used for industrial customers,
- the COM rates can only be used for commercial customers,
- the LAMP rates can only be used for lamp customers

Then you might want to set up a new SA type to indicate such. The advantage of doing this is that you get more control over which rates can be used on a given SA type. The disadvantage is that you proliferate SA types. Unfortunately, it's really a question of taste.

CIS Division/ SA Type	Service Type	Distribution Code	Revenue Class	Rates
CA/G-RES	Gas service	A/R-UTIL	R - residential	GALL-1
CA/G-COM	Gas service	A/R-UTIL	C - commercial	GALL-1
CA/G-IND	Gas service	A/R-UTIL	I - industrial	GALL-1
CA/W	Water service	A/R-UTIL	N/A	WALL-1
CA/E-RES	Electric service	A/R-UTIL	N/A	ERES-1, ERES-2
CA/E-COM	Electric service	A/R-UTIL	N/A	ECOM-1
CA/E-IND	Electric service	A/R-UTIL	N/A	EIND-1
CA/LAMP	Electric service	A/R-UTIL	N/A	ELAMP-1
CA/WW	Waste water service	A/R-UTIL	N/A	WWALL-1
CA/CABLE	Cable service	A/R-UTIL	N/A	CABLE

WARNING:
Don't be too specific in your analysis in respect of rate segmentation because you could end up with a separate SA type for every rate (and you don't want this). We understand this is a very subjective warning, but we recommend that you start out with broad bands of rates that can be used on a SA type and narrow it down if you end up unhappy with the results. For example, you don't have to set up a separate SA type for low-income residential gas customers just because they have a special rate. Rather, you can leave your SA types as they are and treat low-income gas customers as a subset of your residential gas customers. **For more information**, refer to Setting Up Start Options.

NOTE:
Rate override. If a service agreement of this type may be linked to a terms of service record, you must also consider whether or not the rate schedule could be overridden by a template SA linked to the terms of service record. Refer to SA Type - Rate for information about the possible values for this field.

Service Point (SP) Type Segmentation

Every SA Type whose service agreements exist to bill for service point-oriented service has one or more SP types. These define which type of service points can be linked to the SA type's service agreements. If different service agreements have different valid SP types, you will need to split the SA types accordingly.

For each service point-oriented SA type, determine if there are any restrictions in respect of the types of service points that can use the SA type's rates. For example, if we assume that only commercial SP types can be used by commercial customers, industrial SP types by industrial customers, residential SP types by residential customers, your SA types will be:

CIS Division/ SA Type	Service Type	Distribution Code	Revenue Class	Rates	SP Type
CA/G-RES	Gas service	A/R-UTIL	R - residential	GALL-1	G-RES
CA/G-COM	Gas service	A/R-UTIL	C - commercial	GALL-1	G-COM
CA/G-IND	Gas service	A/R-UTIL	I - industrial	GALL-1	G-IND
CA/W-RES	Water service	A/R-UTIL	N/A	WALL-1	W-RES
CA/W-COM	Water service	A/R-UTIL	N/A	WALL-1	W-COM
CA/W-IND	Water service	A/R-UTIL	N/A	WALL-1	W-IND
CA/E-RES	Electric service	A/R-UTIL	N/A	ERES-1, ERES-2	E-RES
CA/E-COM	Electric service	A/R-UTIL	N/A	ECOM-1	E-COM
CA/E-IND	Electric service	A/R-UTIL	N/A	EIND-1	E-IND
CA/WW-RES	Wastewater service	A/R-UTIL	N/A	WWALL-1	W-RES, WW-RES
CA/WW-COM	Wastewater service	A/R-UTIL	N/A	WWALL-1	W-COM, WW-COM
CA/WW-IND	Wastewater service	A/R-UTIL	N/A	WWALL-1	W-IND, WW-IND
CA/CABLE	Cable service	A/R-UTIL	N/A	CABLE	CABLE

Notice the wastewater SA types reference both water and waste water service points. This is intentional as wastewater service uses the consumption from the water service to calculate some part of the wastewater charge.

Company Usage Segmentation

Up to now, we've discussed SA types associated with service agreements linked to your customers. The system has also been designed to keep track of the expenses associated with your company's use of power. If you want the system to do this, you must create at least one SA type for each service consumed by your organization.

For example, if we assume your organization consumes electric, gas, and water service; your SA types will now be as follows:

CIS Division/ SA Type	Service Type	Distribution Code	Revenue Class	Rates	SP Type	Bill Seg Type
CA/G-RES	Gas service	A/R-UTIL	R - residential	GALL-1	G-RES	BD-RATED
CA/G-COM	Gas service	A/R-UTIL	C - commercial	GALL-1	G-COM	BD-RATED
CA/G-IND	Gas service	A/R-UTIL	I - industrial	GALL-1	G-IND	BD-RATED
CA/W-RES	Water service	A/R-UTIL	N/A	WALL-1	W-RES	BD-RATED
CA/W-COM	Water service	A/R-UTIL	N/A	WALL-1	W-COM	BD-RATED
CA/W-IND	Water service	A/R-UTIL	N/A	WALL-1	W-IND	BD-RATED
CA/E-RES	Electric service	A/R-UTIL	N/A	ERES-1, ERES-2	E-RES	BD-RATED
CA/E-COM	Electric service	A/R-UTIL	N/A	ECOM-1	E-COM	BD-RATED
CA/E-IND	Electric service	A/R-UTIL	N/A	EIND-1	E-IND	BD-RATED
CA/WW-RES	Wastewater service	A/R-UTIL	N/A	WWALL-1	W-RES, WW-RES	BD-RATED

CA/WW-COM	Wastewater service	A/R-UTIL	N/A	WWALL-1	W-COM, WW-COM	BD-RATED
CA/WW-IND	Wastewater service	A/R-UTIL	N/A	WWALL-1	W-IND, WW-IND	BD-RATED
CA/CABLE	Cable service	A/R-UTIL	N/A	CABLE	CABLE	BD-RATED
CA/E-COY	Electric service	EXP-COMP	N/A	E CO USE	E-CO USE	COMPUSAG
CA/G-COY	Gas service	EXP-COMP	N/A	G CO USE	G-CO USE	COMPUSAG
CA/W-COY	Water service	EXP-COMP	N/A	W CO USE	W-CO USE	COMPUSAG

Notice the three company usage SA types do not reference an A/R account as their distribution code. This is because when bill segments are created for these types of service agreements, the system must debit an expense account (or contra-revenue account) rather than a receivable account.

Also notice we introduced a new column - Bill Segment Type. Notice that the customer-oriented SA types use the BD-RATED bill segment type and the company usage SA types use the COMPUSAG bill segment type. Different bill segment types are necessary because company usage SA types use a different algorithm to calculate their bill segment's financial transaction algorithm because they don't affect either payoff or current balance.

FASTPATH:
For more information, refer to Designing and Defining Bill Segment Types.

Debt Class Segmentation

Every SA Type has a debt class. The debt class is used to categorize a service agreement's debt for the purpose of credit and collections (C&C) analysis. If a given SA Type has different categories of debt from C&C's perspective, you will have to split the SA Type.

FASTPATH:
For more information about debt class, refer to Designing Your Collection Procedures.

If we assume that your residential services are regulated and your commercial and industrial services are deregulated, we won't have to introduce additional SA types.

CIS Division/ SA Type	Distribution Code	Revenue Class	Rates	SP Type	Bill Seg Type	Debt Class
CA/G-RES	A/R-UTIL	R - residential	GALL-1	G-RES	BD-RATED	REGU
CA/G-COM	A/R-UTIL	C - commercial	GALL-1	G-COM	BD-RATED	UNRE
CA/G-IND	A/R-UTIL	I - industrial	GALL-1	G-IND	BD-RATED	UNRE
CA/W-RES	A/R-UTIL	N/A	WALL-1	W-RES	BD-RATED	REGU
CA/W-COM	A/R-UTIL	N/A	WALL-1	W-COM	BD-RATED	UNRE
CA/W-IND	A/R-UTIL	N/A	WALL-1	W-IND	BD-RATED	UNRE
CA/E-RES	A/R-UTIL	N/A	ERES-1, ERES-2	E-RES	BD-RATED	REGU
CA/E-COM	A/R-UTIL	N/A	ECOM-1	E-COM	BD-RATED	UNRE

CA/E-IND	A/R-UTIL	N/A	EIND-1	E-IND	BD-RATED	UNRE
CA/WW-RES	A/R-UTIL	N/A	WWALL-1	W-RES, WW-RES	BD-RATED	REGU
CA/WW-COM	A/R-UTIL	N/A	WWALL-1	W-COM, WW-COM	BD-RATED	UNRE
CA/WW-IND	A/R-UTIL	N/A	WWALL-1	W-IND, WW-IND	BD-RATED	UNRE
CA/CABLE	A/R-UTIL	N/A	CABLE	CABLE	BD-RATED	UNRE
CA/E-COY	EXP-COMP	N/A	E CO USE	E-CO USE	COMPUSAG	No debt
CA/G-COY	EXP-COMP	N/A	G CO USE	G-CO USE	COMPUSAG	No debt
CA/W-COY	EXP-COMP	N/A	W CO USE	W-CO USE	COMPUSAG	No debt

Notice the three company usage SA types do not have a debt class. This is because their bill segment type's FT algorithm doesn't cause debt to be created and therefore there is no reason to have a debt class. However, you'll need to create a "dummy" debt class - call it N/A - for these SA types because every SA type must reference a debt class.

Budget Billing Segmentation

Many utilities offer their customers levelized payment plans to smooth out the seasonal bill variations. We call this levelized amount the *budget amount*.

- **FASTPATH:**
  Refer to Budget Billing for more information about budgets in general. Refer to Billing - Current Balance versus Payoff Balance for an example of budget billing accounting.

If we assume that you only allow budget billing on the electric and gas residential services, then you'll need to update your CA/G-RES and CA/E-RES SA types:

CIS Division/ SA Type	Dist Code	Rev Class	Rates	SP Type	Bill Seg Type	Debt Class	Recurring Charge Control Info
CA/G-RES	A/R-UTIL	R	GALL-1	G-RES	BD-RATED	REGU	Amount to bill is Not Allowed Amount is Optional Frequency is Monthly Recurring Amount Label is Budget Amount:
CA/G-COM	A/R-UTIL	C	GALL-1	G-COM	BD-RATED	UNRE	
CA/G-IND	A/R-UTIL	I	GALL-1	G-IND	BD-RATED	UNRE	
CA/W-RES	A/R-UTIL	N/A	WALL-1	W-RES	BD-RATED	REGU	
CA/W-COM	A/R-UTIL	N/A	WALL-1	W-COM	BD-RATED	UNRE	

CA/W-IND	A/R-UTIL	N/A	WALL-1	W-IND	BD-RATED	UNRE	
CA/E-RES	A/R-UTIL	N/A	ERES-1, ERES-2	E-RES	BD-RATED	REGU	Amount to bill is Not Allowed Amount is Optional Frequency is Monthly Recurring Amount Label is Budget Amount:
CA/E-COM	A/R-UTIL	N/A	ECOM-1	E-COM	BD-RATED	UNRE	
CA/E-IND	A/R-UTIL	N/A	EIND-1	E-IND	BD-RATED	UNRE	
CA/WW-RES	A/R-UTIL	N/A	WWALL-1	W-RES, WW-RES	BD-RATED	REGU	
CA/WW-COM	A/R-UTIL	N/A	WWALL-1	W-COM, WW-COM	BD-RATED	UNRE	
CA/WW-IND	A/R-UTIL	N/A	WWALL-1	W-IND, WW-IND	BD-RATED	UNRE	
CA/CABLE	A/R-UTIL	N/A	CABLE	CABLE	BD-RATED	UNRE	
CA/E-COY	EXP-COMP	N/A	E CO USE	E-CO USE	COMPUSAG	No debt	
CA/G-COY	EXP-COMP	N/A	G CO USE	G-CO USE	COMPUSAG	No debt	
CA/W-COY	EXP-COMP	N/A	W CO USE	W-CO USE	COMPUSAG	No debt	

Notice the following:

- We updated the two SA types to allow recurring charge information.
- The Recurring Charge Amount is Optional. Why? Because those customers who aren't on a budget plan won't have a recurring charge amount. Those customers on a budget will have a recurring charge amount.

NOTE:

Turn on Eligible for Budget. Besides indicating that these SA types use recurring charges, you also need to turn on the Eligible for Budget switch on the SA type to indicate that this type of SA participates in budget processing.

Override Budget Eligibility. You may plug in an override budget eligibility algorithm on an SA type that is configured to be Eligible for Budget if certain service agreements of this type are not eligible.

FASTPATH:

Refer to the Description of Page under SA Type - Billing for the definition of the recurring charge attributes.

Designing SA Types For SAs Without Service Points

The topics in this section provide guidelines describing how to design the SA types associated with your service agreements that don't have service points.

Overpayment Segmentation

When a customer pays more than they owe, you must decide what to do with the excess money. The following points describe two possibilities:

- You could create a new service agreement to hold the excess (let's call it an overpayment SA). The credit would be transferred from this service agreement to the billable service agreements when the next bill is completed. This means that all billable service agreements have the same opportunity to receive the overpayment when they are billed in the future.
- You could amalgamate the excess payment on one of the existing, billable service agreements. For example, if a customer has both electric and gas service, the excess funds could be kept on either the gas or the electric SA. This would result in the following:
  - The service agreements that do NOT receive the overpayment will have debt when they are next billed.
  - The service agreement that receives the overpayment could have its future debt offset by the overpayment (meaning that it could have a credit balance until the service agreement's future bill segments offset the overpayment amount).
- The above situation is not desirable unless the customer intentionally overpaid one service agreement. The first method (keeping the overpayment on a separate service agreement) obviates this potential problem. Obviously, if your organization sells a single service (and therefore your customers have a single service agreement) you would choose the second method.

You control which method is used by plugging in the appropriate Overpayment Distribution algorithm on each Customer Class (i.e., you can choose a different method for different customer classes). If you choose to hold overpayments on a separate SA, then you must set up an SA Type as described in the following table:

CIS Division/ SA Type	Service Type	Distrib. Code	Eligible for Billing	Debt Class	Pay Seg Type	Do Not Overpay	One-time
CA/OVERPAY	Other	A/P - OVER	Not billed	N/A	Normal	No	Yes

Notice the following about the new overpayment SA type:

- It has an interesting distribution code. This is because when a payment segment is created for this type of service agreements, the system must credit a liability (an overpayment is a liability).
- It's important to indicate that the overpayment SA is a one-time service agreement. Why? Because this means that the system will automatically close the SA when it's balance falls to zero (i.e., when all of the overpayment has been used to satisfy future bills).
- A bill segment type is not needed because the system never creates bill segments for such service agreements (they exist only to hold excess credits).
- You may also want to turn on the alert message
- You must plug-in a bill completion algorithm on this SA type. This bill completion algorithm will transfer the credit balance to the account's other service agreements when the bill is completed. Refer to The Credit Transfer Algorithm for more information about this algorithm.

- You must also reference this overpayment SA type as the parameter value on your overpayment algorithm (this algorithm is plugged in on the desired customer classes). Refer to Overpayment Algorithm for more information about this algorithm.

➤ **NOTE:**
If overpayment means charitable contribution. Some organizations sponsor a program that works as follows - if a customer overpays a bill by a given amount (say $5), this amount is assumed to be a charitable contribution. If you have this requirement, you will create another SA type to hold a customer's charitable contributions. This SA type will look similar to the one described below (see Charitable Contribution Segmentation) except it is not billable. The funds will be credited to this service agreement by creating a new overpayment algorithm that is similar to the base package Overpayment Algorithm. This new algorithm will be very similar to the existing algorithm. The main difference will be that it will have to check if the overpayment amount is an exact value (say $5). If so, it will create a payment segment for the charitable contribution SA type; otherwise it will create a payment segment for the overpayment SA.

Write Off Segmentation

When you write off non-collectable debt, you transfer the receivable from a "normal" service agreement onto one or more write-off service agreements. When the debt is transferred to a write-off service agreement, the distribution code on the "normal" service agreement is credited (typically an A/R GL account), and the distribution code on the write-off service agreement is debited.

You will almost always need a write-off service agreement whose distribution code is the write-off expense. However, you probably don't book all of the write-off amount to a write-off expense account. Why? Because the debt that you're writing off typically contains both revenue and liabilities. At write-off time, you want to book the written off revenue to a write-off expense account and you want to reduce the liabilities (you don't owe the liability if you don't get paid). This means you'll need another SA type for the liabilities. Refer to The Ramifications of Write Offs in the General Ledger for a complete explanation.

The following table contains the minimum number of SA Types that you'll need to hold your write-offs.

CIS Division/ SA Type	Service Type	Distrib. Code	Bill Seg Type	Debt Class	Pay Seg Type	Do Not Overpay
CA/WO-STD	Other	EXP-W/O	Not billed	WO	Normal	Yes
CA/WO-LIA	Other	LIA-General	Not billed	WO	Normal	Yes

Notice the following about the new write-off SA types:

- They have interesting distribution codes. This is because when debt is transferred to these types of service agreements, the system must debit either an expense account (i.e., write-off expense) or a liability account. It's important to note that in The Ramifications of Write Offs in the General Ledger we explain how this liability account may be overwritten with the liability account that was originally booked.
- Neither needs a bill segment type because the system never creates bill segments for such service agreements (they exist only to hold uncollectable debt)
- Even though the debt is not collectable, it still has a debt class. Why? Because the system shows a customer's debt on many inquiries by debt class and it's important to show write-off debt on these queries.
- The combination of Payment Segment Type and Do Not Overpay are important. Refer to The Ramifications of Write Offs in the General Ledger for a complete explanation.

NOTE:
The adjustment type used to set the offending service agreement's current balance equal to its payoff balance is defined on each write-offable SA type. The adjustment type used to transfer the delinquent debt to the write-off service agreement is defined on the write-off SA type.

An Alternative. If you have a limited number of liability accounts, you may decide to have a separate write-off service agreement for each liability account. Doing this would proliferate the number of service agreements created at write-off time. However, it would simplify the remittance of payment to the taxing authority if the reversed liability is ever paid.

Connection Charge Segmentation

If you levy connection charges, you have two options:

- You can create a SA type that exists purely to handle connect charge debt. After doing this, you'd create a start option for this SA type that causes an adjustment to be levied as part of the start service process. This adjustment would contain your standard connection charge. This approach would be used by a utility that had multiple services (e.g., a combined electric, gas, water utility) that only levies a single connection charge regardless of the number of services started. If you use this approach, make sure to indicate the SA type is non billable.
- You can levy a start adjustment on one of your existing SA types (e.g., CA/E-RES). The easiest way to do this is with a start option. On the start option you'd indicate an adjustment to be levied as part of the start service process. This adjustment would contain your standard connection charge. This approach would be used by a utility that had a single service offering (e.g., an electric-only utility). Refer to Setting Up Start Options for more information.

In the table below, we show what would be necessary if you want to have a separate service agreement for the connection charge.

CIS Division/ SA Type	Service Type	Distrib. Code	Bill Seg Type	Debt Class
CA/CONNECT	Other	A/R-UTIL	Not billed	REGU

Notice the following about the new connection charge SA type:

- It has a normal receivable distribution code.
- It doesn't need a bill segment type because the system never creates bill segments for such service agreements (its charged via an adjustment).
- The debt class is interesting - REGU (for regulated). We are intentionally linking the connection charge debt to the same debt class as the regulated debt from which it originates. This way, the C&C process will consider the connection charge debt the same as regulated debt and therefore perform the regulated collection (which results in the severance of all regulated service agreements).

Charitable Contribution Segmentation

If your organization accepts charitable contributions made by your customers, you must create a SA type to hold these contributions.

CIS Division/ SA Type	Service Type	Distrib. Code	Debt Class	Bill Seg Type	Recurring Charge Control Info
CA/CHARITY	Other	A/P-CHAR	CHAR	RECUR	Amount to bill is Not Allowed Amount is Required Frequency is Monthly Recurring Amount Label is Contrib. Amount

Notice the following about the new charitable contribution SA type:

- It has an interesting distribution code. This is because when a payment is distributed to these types of service agreements, the system must credit a payable account (i.e., charitable contribution payable) rather than a receivable account. Note well, we have assumed a receivable is not incurred when the bill segment for the charitable contribution is created.
- It uses an interesting bill segment type - RECUR. This bill segment type was set up to create recurring charges that don't automatically stop at some point in time.
- The debt class is interesting - CHAR (for charity). This is done so that past due charitable contribution debt is treated separately from other types of debt.
- The recurring charge control information is set up as defined.

- **FASTPATH:**
  Refer to the Description of Page under SA Type - Billing**for the definition** of the recurring charge attributes. Refer to Start Option Considerations For SA Types That Use Recurring Charges**for how** you can use start options to automatically populate a service agreement's recurring charge fields with appropriate values when service is started.

WARNING:
It's important that you assign the charitable contribution SA type with a payment segment type that only affect current balance (as opposed to affecting current AND payoff balance). This is because there is no receivable recognized when the contribution is billed and therefore there is no payoff balance to relieve when it's paid.

Payment Arrangement Segmentation

If your organization allows customers to payoff outstanding debt using payment arrangements (e.g., current bill plus $X), you will need a new SA type for every debt class that can have a payment arrangement. If we assume you can have payment arrangements for both regulated and unregulated debt, then you'll need at least two more SA types (you may have more SA types if you need to segregate the payment arrangement receivable amount by utility type (or some other type)).

CIS Division/ SA Type	Service Type	Distrib. Code	Debt Class	Bill Seg Type	Recurring Charge Control Info
CA/PA-REGU	Other	A/R-ARRG	REGU	RECUR-AS	Amount to bill is Not Allowed

					Amount is Required Frequency is Monthly Recurring Amount Label is Arrange Amount
CA/PA-UNRE	Other	A/R-ARRG	UNRE	RECUR-AS	Amount to bill is Not Allowed Amount is Required Frequency is Monthly Recurring Amount Label is Arrange Amount

Notice the following about the new payment arrangement SA types:

- They have an interesting distribution code. This is because when funds are transferred to these types of service agreements, the system must debit a receivable (i.e., payment arrangement receivable).
- They use an interesting bill segment type - RECUR-AS. This bill segment type was set up to create recurring charges that stop when the customer no longer has a payoff balance.
- Each new SA type references the debt class whose debt it will pay off. We are intentionally linking the payment arrangement debt to the same debt class as the regulated debt from which it originates. This way, the C&C process will consider the arrangement debt as the same as regulated debt and therefore perform the regulated collection (which results in the severance of all regulated service agreements).
- The recurring charge control information is set up as defined.

FASTPATH:
Refer to the Description of Page under SA Type - Billing**for the definition** of the recurring charge attributes. Refer to Start Option Considerations For SA Types That Use Recurring Charges**for how** you can use Start Options to automatically populate a service agreement's recurring charge fields with appropriate values when service is started.

Merchandise Segmentation - Installment Billing

If your organization allows customers to purchase merchandise using an installment plan, you must create a SA type for this.

NOTE:
No installments. If the customer must pay for the merchandise in one lump amount, you'd create an SA type similar to the Connection Charge Segmentation example.

CIS Division/ SA Type	Service Type	Distrib. Code	Debt Class	Bill Seg Type	Recurring Charge Control Info

CA/MERCH-I	Merch	A/R-MRCH	UNRE	RECUR-AS	Amount to bill is Not Allowed Amount is Required Frequency is Monthly Recurring Amount Label is Install Amount

Notice the following about the new merchandise SA type:

- It has a normal receivable distribution code.
- It uses an interesting bill segment type - RECUR-AS. This bill segment type was set up to create recurring charges that stop when the customer no longer has a payoff balance.
- The recurring charge control information is set up as defined.

FASTPATH:
Refer to the Description of Page under SA Type - Billing**for the definition** of the recurring charge attributes. Refer to Start Option Considerations For SA Types That Use Recurring Charges**for how** you can use start options to automatically populate a service agreement's recurring charge fields with appropriate values when service is started. Refer to Start Option Considerations For SA Types That Use Initial Adjustments**for how** you can use start options to automatically populate a service agreement's recurring charge fields with appropriate values when service is started.

Deposit Segmentation - Installment Billing

If your organization allows customers to pay deposits using an installment plan, you must create an SA type for this.

NOTE:
No installments. If the customer must pay for the deposit in one lump amount, you'd create an SA type similar to the Connection Charge Segmentation example. Just make sure the adjustment that's levied to charge for the deposit amount doesn't affect payoff balance (when you bill a deposit, the customer doesn't really owe anything because it's not a true receivable from an accountant's perspective).

CIS Division/ SA Type	Service Type	Distrib. Code	Debt Class	Bill Seg Type	Recurring Charge Control Info
CA/DEP-I	Other	A/P-DEPO	DEP	RECURATB	Amount to bill is Required Amount is Required Frequency is Monthly

Recurring Amount Label is Install Amount

Notice the following about the new deposit SA type:

- It has an interesting distribution code. This is because when a payment is distributed to these types of service agreements, the system must credit a payable account (i.e., deposit payable) rather than a receivable account. Note well, we have assumed a receivable is not incurred when the bill segment for the deposit is created.
- It uses an interesting bill segment type - RECURATB. This bill segment type was set up to create recurring charges that stop when the system has billed the Total Amount to Bill.
- The debt class is interesting - DEP (for deposit). This is done so that past due deposit "debt" is treated separately from other types of debt.
- The recurring charge control information is set up as defined. Note well, the Amount to bill is Required.

FASTPATH:
Refer to the Description of Page under SA Type - Billing**for the definition** of the recurring charge attributes. Refer to Start Option Considerations For SA Types That Use Recurring Charges**for how** you can use start options to automatically populate a service agreement's recurring charge fields with appropriate values when service is started.

NOTE:
Bill messages on receipt of deposit in full. The base package includes a special FT Freeze algorithm that can be specified on deposit SA Types. It recognizes when a deposit has been paid in full, and creates a bill messages to inform the customer. Refer to algorithm DEP PIF MSG in Algorithm Types for more information.

Billable Charge Segmentation

You create a billable charge whenever a customer should be charged for a service that occurs outside the normal course of business. For example, you would create a billable charge to charge a contractor for the repair of a ruptured gas line. You can also use billable charges to "pass through" other bill ready charges generated outside the system, by another application, or by a 3rd party supplier.

A billable charge must reference a service agreement. This service agreement behaves just like any other service agreement:

- **Bill segments are created for the service agreement.** Whenever billing is performed for an account with billable charge service agreements, the system creates a bill segment for each service agreement with unbilled charges. If multiple unbilled charges exist for a given service agreement, only one bill segment will be created and it will contain details about all of the billable charges.
- **Payments are distributed to the service agreement.** Payments made by an account are distributed to its billable charge service agreements just like any other service agreement.
- **Overdue debt is monitored.** The credit and collections process monitors billable charge service agreements for overdue debt and responds accordingly when overdue debt is detected.

Therefore, you must set up at least one SA type to hold your billable charge debt. You may have multiple charges based on billing frequencies, A/R booking, debt monitoring, etc. It's really up to you.

The easiest way to determine how many billable charge SA types you'll need is to define every conceivable billable charge (which you should have done when you designed your billable charge templates). Then ask yourself if they have the same billing and payment behavior, if so, you'll have one SA type. If not, you'll need one SA type for each combination.

We will assume your billable charges are all used to levy unusual one-off charges that can be collected in the same way, therefore we'll need one SA type.

CIS Division/ SA Type	Service Type	Distrib Code	Debt Class	Bill Seg Type	Billable Charge Templates	Rate
CA/ONETIME	Other	A/R-UTIL	UNRE	BILLCHRG	TREETRIM DAMAGE	None
CA/PASSTHRU	Electricity	A/R-OTHER	UNRE	BILLCHRG	None	None
CA/ADDON	Electricity	A/R-OTHER	EXTERNAL	BILLCHRG	None	TAXES

Notice the following about the new one time SA type:

- It has a normal receivable distribution code.
- Its debt class is unregulated.
- It uses an interesting bill segment type - BILLCHRG. This bill segment type was set up to create bill segments using billable charges.
- It references the valid billable charge templates that can be used on this SA type.

NOTE:
One Time Charge. The ONE TIME example shown above implies this SA type exists to hold one-time charges. Because of this, you should turn on the One Time Charge switch on the SA type so that service agreement's of this type are automatically closed when final payment is received. You don't have to do this because a customer could have a single billable charge service agreement that is perpetually active for pass through charges (i.e., it doesn't have a stop date). If you do this, the system will create a bill segment for this service agreement whenever it finds an unbilled billable charge linked to the service agreement.

Notice the following about the pass through SA Type:

- It doesn't use the normal distribution code or debt class. This is done so that the debt and receivable can be tracked separately. If these charges were being pass through from another system, you might want to track these financial values separately.
- It still uses the normal bill segment type - BILLCHRG. From a billing perspective, there is no difference between this and the one time SA Type.
- Templates are not relevant - these charges on not created on-line using templates, but are loaded via the Billable Charge Upload Staging.

Notice the following about the add on charges SA Type:

- This is an example of bill-ready charges (similar to pass through) to which the system adds on other charges, for example, taxes.
- It still uses the normal bill segment type - BILLCHRG. From a billing perspective, there is no difference between this and the one time SA Type.
- It also uses a Rate. In this case, the bill creation algorithm (specified on the bill segment type) will take any billable charge lines and attach them to a bill. In addition, these billable charges will include billable charge service quantities (SQs). These service quantities will also be swept onto the bill segment, and the Rate (TAXES in this example) will be

applied. In order for taxes to be calculated, the billable charge SQs must include the total taxable amount - the system is not able to apply the rate on top of the other billable charges. But, it can apply the tax rate to the SQs that are supplied.

- You can also use this technique to bill other rate-ready service quantities, like kWh, CCF, *etc.* This is a way to process rate-ready data for which you have a contract, but you do not know the meter (and therefore, cannot collect real meter reads).
- If the rate has pre-processing calculation groups, these will be applied as well.

FASTPATH:
For more information about billable charge templates, refer to Setting Up Billable Charge Templates.

Over/Under Cash Drawer Segmentation

In order to balance a tender control that is out-of-balance, your organization must set up an account with a service agreement whose SA type references the over/under expense account. You will probably only have one service agreement that references this SA type, but you still must have it if you remit funds via a cash drawer.

FASTPATH:
For more information about over/under processing, refer to How To Get An Unbalanced Tender Control In Balance (Fixing Over/Under).

CIS Division/ SA Type	Service Type	Distrib. Code	Debt Class	Bill Seg Type
CA/OVR UNDR	Other	EXP-OV/UND	N/A	Not billed

Notice the following about the new SA type:

- It has an interesting distribution code. This is because when a payment segment is applied to this type of service agreement, the system must debit an expense account for under amounts (and credit it for over amounts).
- It doesn't need a bill segment type because the system never creates bill segments for such service agreements (it only has over/under payment segments linked to it).
- It uses the N/A debt class because the credit and collections process should never consider debt associated with service agreements of this type (because it's not really debt).

Payment Upload Error Segmentation

If the payment upload process detects an invalid account on a payment upload record, it will create a payment for the suspense service agreement defined on the upload process' tender source (see Setting Up Tender Sources). You should create a special SA type for this service agreement.

FASTPATH:

For more information about the payment upload process, refer to Phase 3 - Create Payment Events, Tenders, Payments, and Payment Segments.

CIS Division/ SA Type	Service Type	Distrib. Code	Debt Class	Bill Seg Type
CA/SUSPENSE	Other	EXP-MISC	N/A	Not billed

Notice the following about the new SA type:

- It has an interesting distribution code. This code should probably be a suspense account. All payment segments that are created for this service agreement will eventually be transferred to a "real" service agreement and therefore this GL account's balance should be zero when no payments are in suspense.
- It doesn't need a bill segment type because the system never creates bill segments for such service agreements (it only has invalid account payment segments linked to it).
- It uses the N/A debt class because the credit and collections process should never consider debt associated with service agreements of this type (because it's not really debt).

CIAC Segmentation

If your company bills and refunds Contribution In Aid of Construction (CIAC) contracts, you must create one or more SA types. CIAC contracts are typically used to levy charges associated with line extensions. These types of service agreements are different from other service agreements because the initial amount charged is refunded to the original payee when new properties (or extensions) are added to the extension (or when the new properties are subsequently billed for service).

NOTE:
Billable charges are used for the original CIAC service agreement. CIAC SA types are always Billable Charge SA types because an operator must specify the exact amount to charge the contractor using a billable charge. Refer to Billable Charge Segmentation for more information.

The following points describe how CIAC processing is implemented in the system:

- When a new line extension is build, you will create a new service agreement that references your CIAC SA type. This service agreement should be linked to a service point associated with the extension. The easiest way to create this service agreement and link it to the service point is by using the Start Account transaction. It's important to remember to define your CIAC SA type as a valid SA type for the SP type used to represent the line extension.
- When a new premise / service point is built that "hangs" off the original extension, the new service point should be linked to the original service point. This can be done by referencing the original service point as the "parent" service point of the new service point using a foreign key value characteristic.
- The system periodically monitors the original CIAC service agreement. The process that performs this monitoring is referenced on the original CIAC service agreement's SA type. The exact processing that takes place during this monitoring is up to your organization's specific business requirement. For example, if you refund 10% of every bill produced for the "downstream" service points to the original contractor, you would have logic in your CIAC monitoring process that looks for recent bill segments produced for the "downstream" service agreements and then creates an adjustment for the master service agreement.

NOTE:

CIAC refunds are idiosyncratic. Because CIAC refund processing is idiosyncratic, we do not supply any CIAC monitoring processes in the base package. This is because the likelihood that they could be used is extremely low because of your organization's unique requirements.

Therefore, if your organization performs CIAC processing, you should create a special SA type.

CIS Division/ SA Type	Service Type	Distrib. Code	Debt Class	CIAC Process	Bill Seg Type
CA/CIAC	Other	A/R - CIAC	UNRE	Refund based on percent of future bills	Billable Charge

Loan Segmentation

If you loan money to customers that is recouped using an amortization schedule, you need to set up an SA type for the loan service agreement.

CIS Division/ SA Type	Service Type	Distrib. Code	Loan A/R Distrib. Code	Debt Class	Bill Seg Type
CA/LOAN	Other	A/R - LOAN	A/R - STLN	UNRE	LOAN

FASTPATH:

Refer to Setting Up The System To Enable Loans for more information.

Non-Billed Budget Segmentation

If you allow your customers to pay set amounts at specified intervals (e.g. every two weeks), you need to set up SA types for Non-Billed Budget service agreements. Non-billed budgets are typically used when your company bills on an infrequent basis and you want to provide your customers with a mechanism to make smaller payments more frequently. You may implement two types of Non-Billed Budgets monitored and unmonitored, each type requiring a different SA type. You may also implement different renewal options for Non-Billed Budgets.

CIS Division/ SA Type	Service Type	Distrib. Code	Non Billed Budget Monitoring	Debt Class	Renewal	Bill Seg Type
CA/NBB-MON	Other	A/P - OVPY	Monitored	NBB	Optional	Not billed
CA/NBB-UNMON	Other	A/P - OVPY	Unmonitored	N/A	Optional	Not billed

- **FASTPATH:**
  For more information about monitored and unmonitored Non-Billed Budgets, refer to Defining Non-Billed Budget Options.

Designing SA Type For Other Segmentations

The earlier parts of this discussion described the most common factors that cause the creation of SA Types. However, many obscure factors could cause the introduction of more SA Types. In this section, we explain these more obscure factors.

WARNING:
We strongly recommend not being too pedantic when considering the factors described in this section. If you can only think of a few strange situations that would necessitate a SA type, think carefully before you introduce it. It's better to be a little less than perfect than end up with large number of obscure SA types.

Cash Distribution Segmentation

Every SA Type has a payment segment type. The payment segment type defines the cash account to which the SA type's payments should be booked. If different service agreements have different cash accounts, you will need to split the SA types accordingly.

Adjustment Profile Segmentation

Every SA Type has one or more adjustment profiles. These profiles define the valid adjustment types that can be booked to the SA type's service agreements. If different service agreements within an SA type have different mixtures of valid adjustment types, you must split the SA types accordingly.

Late Payment Charge Segmentation

An option exists on SA Type that causes the system to generate a late payment charge if payment is not received on time. If you don't levy late payment charges on all service agreements, you will need to determine when you do and design your SA types accordingly.

In addition, if you levy late payment charges, the percentage levied and the algorithm that defines the amount of the outstanding balance subject to the charge is defined on the SA type.

Debt Classification Segmentation

Every SA Type has a debt class. The debt class is used to categorize a service agreement's debt for the purpose of credit and collections (C&C) analysis. If a given SA Type has different categories of debt from C&C's perspective, you must split the SA Type.

- **FASTPATH:**
  For more information about debt class, refer to Designing Your Collection Procedures.

➤ **NOTE:**
Write Off Debt Class vs. Normal Debt Class. An SA type references both write off debt class and normal debt class. An SA type's write-off debt class controls the write-off rules imposed on service agreements of a given type. An SA type's normal debt class controls the collection rules imposed on service agreements of a given type. Refer to Different Collection Criteria For Different Customers And Different Debt for more information about collection rules. Refer to Different Write-Off Criteria For Different Customers And Different Debt for more information about write-off rules.

Allow Estimates Segmentation

Every SA Type has a switch that controls whether the system should have Oracle Utilities Meter Data Management estimate consumption if meter reads are missing at billing time. If a given SA Type has different situations when the system should and should not estimate, you will have to split the SA Type.

➤ **NOTE:**
Override. You can override the value of the SA Type's estimation switch on an individual service agreement. This means that if only a few service agreements don't abide by the SA Type's estimation switch, you can change the switch value of these service agreements.

Severance Criteria Segmentation

Every SA Type has severance criteria. The severance criteria define the severance process used to sever service if the customer doesn't pay. You can have multiple severance processes if different conditions warrant a different process. For example, you may have a different severance process if the customer has life support.

If you have a SA Type that requires different severance conditions other than those currently supported, you can make a programmatic change to introduce the additional conditions OR you can split the SA Type.

- **FASTPATH:**
  For more information about severance, refer to Designing Your Severance Procedures.

Deposit Class Segmentation

Every SA Type that exists to hold a cash deposit will reference a deposit class. The deposit class defines the business rules that control various functions including interest calculation and refund criteria. You will need multiple deposit SA Types if any of the deposit class' rules / conditions differ for different types of deposits. For example, if residential customers use a different recommended deposit algorithm as compared to commercial customers, you'd need one SA type for residential deposits and another for commercial deposits (where the residential deposit SA type will reference the residential deposit class and the commercial deposit SA type will reference the commercial deposit class).

You will need additional deposit SA types if your customers can have multiple deposits where each deposit is restricted to a specific type of debt. For example, if separate deposits are held for regulated and unregulated debt (and a customer could hold a combination of regulated and unregulated debt), you'd need one SA type for regulated deposits and another for unregulated deposits.

- **FASTPATH:**
  For more information about deposit class, refer to Designing and Defining Deposit Classes.

Sub SA Types

If you operate in a deregulated environment AND if you provide billing services for other service providers, you will need to create SA types to handle the billing of the service providers' charges.

- **FASTPATH:**
  Refer to Designing Your SA Types And Start Options For Sub SAs for more information.

Financial Settlement SA Types

If you operate in a deregulated environment, you may have to create financial settlement service agreements for the service providers. As explained in Service Providers Have Service Agreements Too, We Bill For Them and They Bill For Us service providers require a service agreement to hold adjustments used to increase how much you owe the service provider (or how much they owe you).

- **FASTPATH:**
  Refer to Designing SA Types For Service Provider Financial Settlements for more information.

Usage Request SA Types

Oracle Utilities Meter Data Management stores meter reading information and is responsible for calculating bill determinants during billing. The SA types for these customers will require special setup.

- **FASTPATH:**
  For more information about usage requests, refer to The Big Picture Of Usage Requests.

Initial Consumption Period Considerations

Bill segments produced for a service agreement have two time periods:

- The bill segment period. The bill segment period defines the entire period of time covered by a bill segment's charges.
- The consumption period. The consumption period defines the period of time used to calculate the number of days for daily charges.

The consumption period almost always starts one day after the bill segment period. The consumption period always ends on the bill segment's end date. For example, a bill segment period that spans 5-Jan-2002 through 6-Feb-2002 will almost always have a corresponding consumption period of 6-Jan-2002 through 6-Feb-2002. The reason that the start dates don't match is because a bill segment's start date equals the end date of the prior bill segment (i.e., the start date was already counted in the previous bill segment's consumption period and we don't want to count it twice).

The only time when the previous paragraph isn't true is the first bill segment that's produced for a new service agreement. This is because different utilities count the first day of a new service agreement differently than others. Because of this, a flag exists on SA Type called **Initial Start Date Option**. This flag controls whether the service agreement's start date is included in the consumption period in a service agreement's first bill segment.

The following table describes the ramifications of the options you can set for this flag.

Flag Value	Consumption Period Calculation	Use This Option When
Add 1 Day Always	The consumption period's start date is calculated by adding 1 day to the service agreement's start date. (The SA start date is never included in the consumption period for the first bill segment.)	You want the initial bill and all subsequent bills to have a consistently calculated consumption period (i.e., the consumption period for the first and all other bills is always one day less than the bill segment's period).
Add 1 Day for Back-to-back	A back-to-back occurs when any service point for this service agreement was previously linked to a different service agreement that was stopped on the same date that the new service was started (i.e., there is no gap in the service). If a back-to-back is detected, the consumption period start date is calculated by adding 1 day to the SA start date. If no back-to-back is detected, the start date of the consumption period is the SA start date.	You want to flexibly handle consumption period calculation. If you start customers on the same date as the stop date of the previous customer, billing does not include the SA start date in the consumption period. However, if you start a new customer one day (or more) after the stop date of the previous customer, billing includes the first day of the service agreement in the consumption period.
Include First Day	The start date of the consumption period is the service agreement start date. (The SA start date is always included in the consumption period.)	You want to always include the first day AND you will never encounter a back-to-back situation.

The example below shows how the consumption period would be calculated with the various options for a customer who starts service on January 1.

	First Bill **Meter Read: Jan 31** **Bill Segment Period: Jan 1 to Jan 31**	Second Bill **Meter Read: Feb 28** **Bill Segment Period: Jan 31 to Feb 28**	Third Bill **Meter Read: Mar 31** **Bill Segment Period: Feb 28 to Mar 31**

Consumption period using Add 1 Day Always	Jan 2 to Jan 31 (**30 days**)	Feb 1 to Feb 28 (28 days)	Mar 1 to March 31 (31 days)
Consumption period using Add 1 Day for Back-to-back when back-to-back is detected	Jan 2 to Jan 31 (**30 days**)	Feb 1 to Feb 28 (28 days)	Mar 1 to March 31 (31 days)
Consumption period using Add 1 Day for Back-to-back when back-to-back is NOT detected	Jan 1 to Jan 31 (**31 days**)	Feb 1 to Feb 28 (28 days)	Mar 1 to March 31 (31 days)
Consumption period using Include First Day	Jan 1 to Jan 31 (**31 days**)	Feb 1 to Feb 28 (28 days)	Mar 1 to March 31 (31 days)

There may be SA types for which the value of this flag does not affect the consumption period calculation and still other SA types where this flag is never used. For example,

- For billable charge service agreements, the consumption period is equal to the start and end dates on the billable charge and therefore this flag is not applicable.
- A sub SA either inherits the consumption period from the master SA or it uses billable charges. As a result, billing does not use this flag.
- For some service agreements, the charges on the rate are not affected by the consumption period. For example, if you have a customer with a simple meter and a simple usage-based charge, billing calculates the amount of consumption between the start reading and end reading and applies the rate (i.e., the number of days in the consumption period doesn't impact the charges in some rates).
- Some service agreements are not billed, for example, overpayment service agreements. For this type of service agreement and other types of service agreements that are not billed, this flag is not applicable. However, the system does not prevent a value from being entered in these cases to allow for an implementation process to use the flag if needed.

Setting this flag to an appropriate value is significant for certain types of service agreements.

- For services whose rate includes daily charges, the configuration of this flag may impact the first bill segment for the service agreement. For example, if the first bill period is October 1st through October 31st, do you consider consumption period to be 31 days or 30 days? How you want to bill the customer on the first bill determines how you set this flag.
- For service agreements that request usage (or bill determinants) based on interval data, the interval usage recordings typically begin on the first day of service. If you add 1 to the start date, you miss billing for intervals on the first day. In this case, you should set the value to Add 1 Day for Back-to-back.

Processing Sequence Considerations

You may have customers with a complex collection of contracts such that the calculation for one bill segment relies on information calculated by another bill segment for the same account. For example, perhaps you need to process your bill segments as follows:

- Step 1: Calculate bill segments for all the account's "single site" service agreements (i.e., each service agreement related to a single premise)
- Step 2: Calculate the bill segment for additional charges for the "head office" service agreement where the charges are based on the aggregated consumption of all premises from all bill segments (calculated in the first step).

To create the "head office" bill segment for the account, you must control the order in which the system creates the bill segments for each service agreement.

The SA type allows you to indicate a billing processing sequence that controls the order in which the service agreements are processed. The processing sequence is optional and service agreements are processed in ascending order of their SA type's billing processing sequence. As a result, SA Types with a processing sequence of zero are processed first, then 1, then 2, etc.

> **NOTE:**
> If you have sub SAs linked to a master SA, the billing processing sequence is used first to order the creation of bill segments for the master SAs. If you populate a processing sequence on an SA type for a sub SA, it is used to control the order in which the sub SAs for a given master SA are processed relative to each other. Refer to Designing Your SA Types And Start Options For Sub SAs for more information.

The billing processing sequence also controls the order of service agreements in the following other processes:

- Execution of pre bill completion algorithms. The system processes each service agreement in the billing processing sequence order. Within each service agreement, the pre bill completion algorithms on its SA type are processed in the order of the algorithm's sequence.
- Execution of bill completion algorithms. The system processes each service agreement in the billing processing sequence order. Within each service agreement, the completion algorithms on its SA type are processed in the order of the algorithm's sequence.

Designing Prepayment Billing Options

This section provides guidelines describing how to design service agreements that handle smart meter prepaid billing.

When a customer is subject to prepayment, normal cycle-based billing is not used to calculate the bill segments for their service. Rather, a prepay billing (PPB) process creates bill segments frequently (for example, daily) and periodically requests funds to maintain an adequate payoff balance. Bills will only be used to periodically inform the customer of the various financial transactions that have occurred since the last bill was produced.

Implementing prepayment billing requires the configuration of the following objects:

- SA Type
- Adjustment Types
- Bill Segment Type
- Service Task Type

These configuration tasks are described below.

> **NOTE:** For additional information, see the detailed descriptions in the application for the C1-PrepayBillerTaskType and C1-PrepayBillerTask business objects.

SA Type for Prepay Billing

To configure a prepay billing SA Type, create a new SA type with the following attributes:

- Set the Special Role flag to **Bill Determinants Required.** The Prepay Option should be set to **Prepaid Smart Meter.**
- Specify the SA Creation system event with the algorithm CI_CREATEPPB to create the service agreement's prepay biller.
- Specify the Payment Freeze system event with the algorithm STPZ-RMVC to set current amount to zero on a payment.

- If required, specify the Bill Completion system event with an algorithm of the type C1-CREPPBBM to add a bill message that informs an end customer of their prepaid charges for the period. You may need to create an algorithm of this type if it does not exist in your system.
- **Eligible for Budget** should be set to false.
- Reference a new **Prepay Bill Segment Type**, which is described below.

The following adjustment types must also be defined on the prepay SA type:

- **Adjustment Type (Xfer)** specifies the type of adjustment used to transfer funds from the delinquent service agreements to the prepaid service agreement
- **Adjustment Type (Current=0)** specifies the type of adjustment used to set the prepaid current balance to zero after debt has been transferred
- **NSF Adjustment Type** specifies the type of adjustment used to levy an NSF charge. The NSF Adjustment Type is described in greater detail below.

The new SA types must be added to the data value mapping between CCB and MDM. This is to enable the synchronization of SAs of this type to MDM. The new CCB SA types can be mapped to existing US types in MDM. For additional information about this configuration, please see the integration documentation.

Adjustment Types for Prepay Billing

- **Funds Request Adjustment Type:** This adjustment type captures the type of adjustment used to request more prepaid funds. This adjustment type is specified on the prepaid biller task type. Specify the following algorithms on this adjustment type:
  - **Adj. Financial Algorithm** should reference the Adjustment FT creation algorithm ADJT-CA (Payoff=0/Current=Adj Amount (No GL)).
  - **Adjustment Freeze Algorithm** should reference an algorithm of the type C1-PPBADJFRZ (Set Prepaid Funds Request FT Details). You may need to create an algorithm of this type if it does not exist in your system.
- **NSF Adjustment Type:** For prepaid customers on autopay, when a user cancels an autopay tender the cancel reason must indicate that an NSF charge should be levied. The system invokes the NSF Charge algorithm specified on the tender's account's customer class which creates the NSF adjustment on the appropriate SA. The charge is levied using the NSF Adjustment Type specified on the SA's SA type. This adjustment type must reference the new adjustment freeze algorithm that will disable automatic payments on the PPB. This algorithm must be based on the algorithm type C1-DISPPBAP (Disable PPB's Automatic Payment). You may need to create an algorithm of this type if it does not exist in your system.

Bill Segment Type Required for Prepay Billing

Configure a new bill segment type for prepaid service agreements. This bill segment type should have the following attributes:

- **Create Algorithm** should be set to C1-BSBS-UR to create bill segment using a usage request.
- **Financial Algorithm** should be set to a new algorithm with algorithm type C1-NEM-GL (Payoff Amt = Bill Amt / Current Amt = 0 (GL affected)).You may need to create an algorithm of this type if it does not exist in your system.
- **Get Consumption Algorithm** should be set to C1-BSGC-USG to get bill segment consumption using a usage request.

Service Task Type for Prepay Billing

Configure a new service task type for prepay billing. This task type should reference the Prepay Biller (PPB)Task Type Business Object (C1-PrepayBillerTaskType). The task type must be specified on the new prepay billing SA creation algorithm. The task type captures the adjustment type used when an adjustment is created to request additional funds from the customer. Ensure that this adjustment type references the adjustment freeze algorithm (C1-PPBADJFRZ) that populates the financial transaction's arrears date.

SA Types And The Financial Design

In this section, we provide an example of how our SA Types map to Bill Segment Types, Payment Segment Types, and Adjustment Profiles. This example is meant to help solidify the power of the financial model, it is not necessarily indicative of how your specific implementation will look.

WARNING:
If you are not comfortable with the topics described in Defining Financial Transaction Options, the following table will not make sense.

Division/SA Type	Distribution Code	Bill Segment Type	Payment Segment Type	Adjustment Profiles
CA/G-RES	A/R-UTIL	BD-RATED	NORMAL	BALXFER, MISCFEES, BUDGET
CA/G-COM	A/R-UTIL	BD-RATED	NORMAL	BALXFER, MISCFEES
CA/G-IND	A/R-UTIL	BD-RATED	NORMAL	BALXFER, MISCFEES
CA/W-RES	A/R-UTIL	BD-RATED	NORMAL	BALXFER, MISCFEES
CA/W-COM	A/R-UTIL	BD-RATED	NORMAL	BALXFER, MISCFEES
CA/W-IND	A/R-UTIL	BD-RATED	NORMAL	BALXFER, MISCFEES
CA/E-RES	A/R-UTIL	BD-RATED	NORMAL	BALXFER, MISCFEES, BUDGET
CA/E-COM	A/R-UTIL	BD-RATED	NORMAL	BALXFER, MISCFEES
CA/E-IND	A/R-UTIL	BD-RATED	NORMAL	BALXFER, MISCFEES
CA/WW-RES	A/R-UTIL	BD-RATED	NORMAL	BALXFER, MISCFEES
CA/WW-COM	A/R-UTIL	BD-RATED	NORMAL	BALXFER, MISCFEES
CA/WW-IND	A/R-UTIL	BD-RATED	NORMAL	BALXFER, MISCFEES
CA/CABLE	A/R-UTIL	BD-RATED	NORMAL	BALXFER, MISCFEES
CA/E-COY	E-COMP	COMPUSAG		
CA/G-COY	E-COMP	COMPUSAG		
CA/W-COY	E-COMP	COMPUSAG		
CA/WO-STD			NORMAL	BALXFER
CA/WO-LIA			NORMAL	BALXFER
CA/CONNECT			NORMAL	BALXFER, MISCFEES
CA/CHARITY	A/P-CHAR	RECUR	CHARITY	CHARITY
CA/PA-REGU	A/R-ARRG	RECUR-AS	NORMAL	BALXFER, MISCFEES, DPA
CA/PA-UNRE	A/R-ARRG	RECUR-AS	NORMAL	BALXFER, MISCFEES, DPA
CA/MERCH-I	A/R-MRCH	RECUR-AS	NORMAL	BALXFER, MISCFEES, MERCH

CA/DEP-I	A/P-DEPO	RECUR-AS	NORMAL	BALXFER, MISCFEES, DEPOSIT
CA/ONETIME	A/R-UTIL	BILLCHRG	NORMAL	BALXFER, MISCFEES
CA/OVR UNDR	EXP-OV/UND		NORMAL	
CA/OVERPAY	A/P-OVER		NORMAL	BALXFER
CA/SUSPENSE	A/R-SUSP		NORMAL	
CA/NBB	A/P-OVPY		NORMAL	BALXFER, MISCFEES, NBB

If you operate in a deregulated environment, you will also have additional SA types as described under Designing Your SA Types And Start Options For Sub SAs and Designing SA Types For Service Provider Financial Settlements you may have additional SA types.

Setting Up SA Types

In the previous section, Designing SA Types, we presented a case study that illustrated a mythical organization's SA types. In this section, we explain how to use the windows on the SA Type window group to maintain your SA Types.

SA Type - Main Information

Open **Admin** > **Customer** > **SA Type** > **Add** to define core information about your SA Types.

WARNING:
Every SA Type is owned by a CIS Division. This Division controls many values that can be referenced on the SA Type. If you don't understand Divisions and their place in the application, do NOT attempt to set up your SA Types. Rather, refer to CIS Division Portal before proceeding.

Description of Page

Enter a unique combination of **CIS Division** and **SA Type** for every service agreement type.

Enter a **Description** for the SA type.

Service Type defines the type of service associated with the SA type. If the SA type has rates, only rates belonging to this service type may be linked to the SA type.

Select the **Distribution Code** and **GL Division** that defines the receivable account for receivable-oriented service agreements. For non-receivable oriented service agreements, this distribution code is typically as follows:

- Charitable contributions. The distribution code is a charity payable account.
- Deposits. The distribution code is a deposit payable account.
- Non-Billed Budgets. The distribution code is an overpayment payable account.
- Company usage. The distribution code is a company usage expense account.
- Write off. The distribution code is a write-off expense account.
- Payment arrangements. The distribution code is a payment arrangement receivable account.

- **FASTPATH:**

For more information about GL accounts, refer to The Source Of GL Accounts On Financial Transactions.

Select the **Revenue Class** associated with the SA Type (and its service agreements). The revenue class may affect the revenue account(s) generated by the service agreement's rate.

●

FASTPATH:
Refer to Designing Calculation Groups and Rules for more information about revenue class.

Turn on **Start Options Required** if you want to force a customer service rep to choose a start option when they start service for this SA Type (on the Start Account window). If this switch is off and a rate is required for the SA Type, the system defaults the SA type's default rate on new service agreements. The default rate is defined on the SA Type - Rate page.

●

FASTPATH:
Refer to Setting Up Start Options for more information about the pros and cons of requiring start options.

Select the **Pay Segment Type** that defines how payment segments linked to service agreements of this type affect:

- The service agreement's payoff and current balances

●

FASTPATH:
For more information about payment segment types, refer to Setting Up Payment Segment Types.

When a tender is canceled, a cancellation reason must be supplied. If the cancellation reason indicates a NSF (non sufficient funds) charge should be levied, the system invokes the Levy an NSF Charge algorithm specified on the tender's account's customer class. Because adjustments must be linked to a service agreement, the algorithm must determine the appropriate service agreement to use to levy the adjustment based on business rules. The charge is levied using the **NSF Adjustment Type** of the appropriate service agreement's SA type.

WARNING:
You must specify adjustment type profiles on the SA type (on the Adjustment Type window) before adjustment types will appear in the above drop downs.

●

FASTPATH:
For more information about adjustment types, refer to Setting Up Adjustment Types. For more information about cancellation reasons, refer to Setting Up Payment Cancellation Reasons.

Select the **Payment Priority**. This field is available for use by the algorithms that distribute partial payments amongst an account's service agreements. Higher priority service agreements will have their debt relieved before lower priorities. Refer to Distribution Based on Payment Priority and Delinquent Payment Distribution Algorithm for information about payment distribution algorithms that use this field.

➤

NOTE:

The values for this field are customizable using the Lookup table. This field name is PAY_PRIORITY_FLG.

●

FASTPATH:
For more information about distribution priority, refer to Distributing A Payment Amongst An Account's Service Agreements.

Select the **Delinquent Payment Priority**. This field is available for use by the algorithms that distribute partial payments amongst an account's service agreements. Higher priority service agreements will have their debt relieved before lower priorities. Refer to Delinquent Payment Distribution Algorithm for information about a payment distribution algorithm that uses this field.

➤

NOTE:
The values for this field are customizable using the Lookup table. This field name is DEL_PRIORITY_FLG.

Turn on **Do Not Overpay** if the system is not allowed to distribute an overpayment to this type of service agreement (i.e., the service agreement is not allowed to have a system-created credit balance). This field is available for use by algorithms that distribute overpayments. Refer to Overpayments Held On Highest Priority Service Agreement for information about an overpayment algorithm that uses this field.

Turn on **Late Payment Charge** if the system should generate a late payment charge for this type of service agreement if payment is not received on time. If this is turned on, you must define the **LPC Calc. Algorithm** used to calculate the late payment charge amount. Refer to Defining Late Payment Charge Options for more information about late payment charges. Examples of algorithm types used for calculating late payment charges are BILPC-SPRC and BILPC-TOTAL.

Define the **Adj**ustment **Type (Synch Curr)** that will be used to synchronize (make equal) the current amount with the payoff amount on a service agreement of this type. This type of processing happens as follows:

- Most write-off algorithms that perform financial efforts (e.g., writing off debt), will issue an adjustment of this type if the service agreement's current and payoff balances are not equal.
- If a user stops a customer on a budget plan, the system issues adjustments of this type to synchronize the customer's current and payoff balances.
- If a user stops a service agreement covered by a**Non-Billed Budget**, the system issues adjustments of this type to synchronize the customer's current and payoff balances.
- If a cancellation of a bill segment occurs after a customer has stopped participating in a budget plan, an adjustment of this type is issued to synchronize the imbalance created when the bill segment's financial transaction is canceled.

Turn on **CIAC SA Type** and specify an appropriate **CIAC Refund Process** if service agreements of this type are used to bill for Contribution In Aid of Construction (CIAC) charges. Refer to CIAC Segmentation for more information.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference CI_SA_TYPE.

SA Type - Detail

Open **Admin** > **Customer** > **SA Type** > **Search** and navigate to the **Detail** page to define additional details about a given SA type.

Description of Page

Turn on **Display As Alert** if Control Central should display an alert if an account has a service agreement of this type that isn't Closed or Canceled. If this switch is on, also enter the **Alert Information** to appear on Control Central. We recommend only using this feature on unusual SA types (e.g., payment arrangements, write-offs) so that a CSR is not presented with an alert for every SA type.

If this SA Type is used for any of the **Special Roles**, defined in the dropdown, indicate which one. Valid values are: Billable Charge, Cash Deposit, Loan, Non-Billed Budget, Payment Arrangement, Write Off, Bill Determinants Required. This information is used on windows with functionality that can only be used by service agreements used for specific roles. For example, the Billable Charge window group can only reference Billable Charge service agreements.

If Special Role is Cash Deposit, you must define the **Deposit Class** of the deposit. You should also define a **Deposit Class** on every SA type to which a given deposit can be distributed.

●

FASTPATH:
Refer to What Do Deposit Classes Do? for more information.

If the Special Role is Loan, you must also define the following fields:

- Use the **Interest Bill Factor** to define the bill factor code for the loan interest rate.
- Use **Override Interest Flag** to indicate whether the interest rate defined on the interest bill factor may be overridden at the SA level. If you select Allowed, the interest rate may be overridden by a contract value on a start option or the SA.
- Use the **Loan A/R Distribution Code** to define the distribution code to be used when posting the short-term receivable amount to the general ledger (the normal distribution code is used for the long-term receivable). If the normal distribution code is the same as the **Loan A/R Distribution Code**, the SA type does not differentiate between long- and short-term receivables. If the two distribution codes are different, the SA type differentiates between long- and short-term receivables.

●

FASTPATH:
Refer to Defining Loan Options for more information about **Interest Bill Factor**, **Override Interest Flag** and **Loan A/R Distribution Code**.

If the Special Role is Non-Billed Budget , you must also define the following:

- Use **Adjustment Type (Xfer)** to specify the type of adjustment used to the transfer accumulated credit from the Non-Billed Budget SA to the SAs covered by the Non-Billed Budget when the account is billed or the Non-Billed Budget SA is stopped.
- Use the **Non-Billed Budget Monitoring** to specify whether the Non-Billed Budget is monitored by the account debt monitor.

If the SA type is defined as Eligible for Non-billed Budget, you must also define the following:

- Use **Adjustment Type (Current=0)** to specify the type of adjustment used to set the service agreement's current balance to zero when a service agreement of this type is linked to an active, monitored Non-Billed Budget.

If the Special Role is Payment Arrangement, you must also define two adjustment types:

- Use **Adjustment Type (Xfer)** to specify the type of adjustment used to transfer funds from the delinquent service agreements to the payment arrangement service agreement.
- Use **Adjustment Type (Current=0)** to specify the type of adjustment used to set the payment arrangement's current balance to zero after funds have been transferred.

If the Special Role is Write Off, you must also define the following adjustment types:

- Use **Adjustment Type (Xfer)** to specify the type of adjustment used to transfer funds from the uncollectable service agreements to the write off service agreement.

If the Prepay Option is **Prepaid Smart Meter**, you must also define two adjustment types:

- Use **Adjustment Type (Xfer)** to specify the type of adjustment used to transfer funds from the delinquent service agreements to the prepaid service agreement.
- Use **Adjustment Type (Current=0)** to specify the type of adjustment used to set the prepaid current balance to zero after debt has been transferred.

WARNING:
You must specify adjustment type profiles on the SA type (on the Adjustment Type window) before adjustment types will appear in the above drop downs.

The **Stop Option Flag** can be used to automatically stop a service agreement when all other service agreements of an account are stopped. Valid values are: Automatically Stop SA. When all service point related service agreements of an account are stopped then all additional service agreements where the **Stop Option** is set to Automatically Stop SA will also be stopped. This is useful, for example, with charitable contribution service agreements. The charity service agreement should be stopped when all utility related services are stopped.

- **FASTPATH:**
  Refer to The Lifecycle of a Service Agreement for more information on the **Stop Option.**

Turn on **One Time Charge** if this SA type is used for one-time invoices. When a one-time invoice service agreement is created, the system sets the stop date of the SA to be equal to the start date.

Turn on **Sub SA** if this SA type is used to define the business rules for sub service agreements.

- **FASTPATH:**
  Refer to Sub Service Agreements for more information about sub service agreements.

Renewal of SAs of this type may be Optional, Not Allowed or Required depending on your business processes. If renewal is not allowed, the SA expires on the expiration date. Renewal treatment is an important consideration for SAs that require an expiration date, such as non-billed budget SAs.

If renewal is required or optional, specify the **Days Before Expiration for Renewal**. Note that currently this is only used by Non-Billed Budgets to calculate the renewal date based on the expiration date.

If the Special Role is Non-billed Budget, **Non-Billed Budget Monitoring** must indicate whether the Non-Billed Budget is monitored by the account debt monitor.

- **FASTPATH:**
  Refer to Credit and Collections and Non-Billed Budgets for more information about monitoring Non-Billed Budgets.

Where Used

The alert information is used by Control Central to alert a CSR when unusual service agreements exist for an account. Refer to Control Central - Main for more information.

Only SA types designated as being Billable Charge may have billable charges linked to them. Refer to Maintaining Billable Charges for more information.

Only SA types designated as being Cash Deposit are processed by the various deposit-related background processes (e.g., interest calculation, automatic refund, etc.). Refer to The Big Picture Of Deposits for more information.

Only SA types designated as Loan are used to define the loan terms for a loan SA. Refer to **Loans** for more information.

Only SA types designated as Non-billed Budget may be used to set up Non-Billed Budgets. Refer to **Non-Billed Budgets** for more information.

Only SA types designated as being Payment Arrangement may be used on the payment arrangement window group. Refer to Setting Up Payment Arrangements for more information.

Only SA types designated as being Write Off may be specified as the write off SA type on distribution codes. Refer to Setting Up Distribution Codes for more information.

Only service agreements whose SA type is designated as being Write Off appear on the Write Off SAs query. Refer to Write Off - Write Off SAs for more information.

SA Type - Billing

Open **Admin** > **Customer** > **SA Type** > **Search** and navigate to the **Billing** page to define how the system manages bill segments for service agreements of a given SA type.

Description of Page

Turn on **Eligible for Billing** if the system should create bill segments for service agreements of this type. This will typically be turned on for all service agreements except for those used to hold write-off amounts or to levy one-off adjustments.

Define the minimum number of days a bill segment (other than the final segment) must span using **Minimum Days for Billing**. This is useful to prevent initial bill segments that span only a few days.

- **FASTPATH:**
  For more information about minimum days, refer to Preventing Short Bill Segments.

Select the **Bill Segment Type** that controls both how bill segments for this SA Type will be created and how the related financial transaction affects the general ledger and the customer's debt.

- **FASTPATH:**
  For more information about bill segment types, refer to Setting Up Bill Segment Types.

Use **Default Description on Bill** to define the verbiage that should print on the customer's bill.

➤ **NOTE:**
Rates overwrite this description. The Default Description on Bill is not applicable for service agreements whose charges are calculated using a rate. Why? Because the description that appears on the bill segment is defined on the rate schedule's rate version calculation group.

Billable charges overwrite this description. The Default Description on Bill is not applicable for service agreements whose charges are calculated using a billable charge. Why? Because the description that appears on the bill segment is defined on the billable charge.

Use the **Billing Processing Sequence** if you need to control the order in which service agreements linked to this SA type are processed by billing.

●

FASTPATH:
Refer to Processing Sequence Considerations for more information.

Use **Bill Print Priority** to define the order in which the SA type's bill segments should appear on bills (relative to the other SA types that appear on a bill).

➤

NOTE:
The values for this field are customizable using the Lookup table. This field name is BILL_PRT_PRIO_FLG.

Use **Max Bill Threshold** if you want the system to generate a bill error when a bill segment is produced in batch that exceeds a given value. These bill errors will appear on the standard billing queries and To Do lists. If, after reviewing the high value bill segment, an operator truly intends to send the bill out, they should regenerate the bill. Refer to How To Correct A Bill Segment That's In Error for more information.

WARNING:
The value entered in this field will DEFAULT onto service agreements of this type when they are first created. An operator may change the default value on a service agreement in case a specific customer has unusually high bills that continually error out. It's important to be aware that if you change the value of High Bill Amount on an SA type and there already exist service agreements of this type, the existing service agreements will contain the original value (the new value on the SA type will not be propagated on the existing service agreements).

Use **Graph Unit Of Measure** to define the unit of measure of the graphed consumption on the bill (if any).

Turn on **Allow Estimates** if the system should indicate to Oracle Utilities Meter Data Management whether to generate estimated consumption if meter reading(s) cannot be found at billing time. This value is defaulted onto service agreements and can be overridden on an individual service agreement.

Turn on **Characteristic Premise Required** if a characteristic premise must be linked to the service agreement when the service agreement is activated. The characteristic premise is used to define the taxing authorities associated with the service agreement's bill segments. It is also used to identify where the service agreement's service is located on various windows.

●

FASTPATH:
For more information about how characteristic premise is used, refer to**An Illustration Of A Bill Factor And Its Characteristics**.

Use the **Initial Start Date Option** to control how billing should calculate the consumption period for the very first bill for service agreements of this type. This field is not applicable for sub SA types or SA types with a special role of

Billable Charge. Valid values are Include First Day, Add 1 Day Always and Add 1 Day for Back-to-back. Refer to Initial Consumption Period Considerations for more information.

Non-metered service agreements may have the end date of their bill segments defined on a user-maintained bill period schedule. This option is used when bill segments must fall on strict calendar boundaries (e.g., quarterly bills that end on the last day of the quarter). If this SA type should be billed like this, select Use Bill Period in the **Use Calendar Billing** field. When this option is used, you must define the **Bill Period** whose schedule defines the bill segment end dates.

- **FASTPATH:**
  For more information about bill period schedules, refer to Designing Bill Periods. For more information about other bill end date methods, refer to Ways To Control The End Date Of A Bill Segment.

Instead of the Use Bill Period method, non-metered service agreements may have their bill segment end date based on the first day of service. For example, if service started on the 16th of some month, the ongoing bill segments will start on roughly the 16th of each month. This option is frequently used to bill for garbage or cable service. If this SA type should be billed like this, select Anniversary Future Billing or Anniversary Past Billing in the **Use Calendar Billing** field. When either option is used, you must define the **AnniversaryBill Frequency**. This frequency defines the amount of time between bill segments.

- **FASTPATH:**
  For more information about anniversary billing, refer to Using The Anniversary Method. For more information about other bill end date methods, refer to Ways To Control The End Date Of A Bill Segment.

Total Bill Amount indicates whether service agreements of this type can use the total amount to bill field on the service agreement page. Valid values are Not Allowed and Required. Only SA types used to bill for deposits or loans should have this field set to Required.

If Required is selected, you must enter the **Total Amount To Bill Label**. The **Total Amount To Bill Label** defines the label that prefixes the total bill amount on the service agreement page for service agreements of this SA type.

- **FASTPATH:**
  For more information about total amount to bill and deposit service agreements, refer to Total Amount To Bill. For more information about total amount to bill and loan service agreements, refer to**Setting Up The System To Enable Loans**.

Recurring Charge indicates whether service agreements of this type can use the recurring charge field on the service agreement window. Valid values are Not Allowed, Optional and Required. If either Optional or Required are used, you must enter:

- **Recurring Chg Amt Label**. This defines the label that prefixes the recurring charge amount on the service agreement window for service agreements of this SA type.
- **Recurring Charge Frequency**. This defines the following:
  - Specifies the frequency at which the Recurring Charge Amount specified on service agreements of the SA Type is to be billed.
  - Serves as the basis for proration of the Recurring Charge Amount.
  - Specifies the frequency at which service agreements of the SA Type without a rate and/or meters will be billed.

- **FASTPATH:**
  For more information about how to use the recurring charge information, refer to Charitable Contribution Segmentation, Merchandise Segmentation - Installment Billing, Deposit Segmentation - Installment Billing, Payment Arrangement Segmentation, Budget Billing Segmentation, The Terms Of A Loan Are Stored On A Service Agreement.

Turn on **Eligible for Budget** if service agreements of this type can participate in budget billing. If this switch is turned on, then you must define the **Adj**ustment **Type (Synch Current)** that will be used to synchronize (make equal) the current amount with the payoff amount on a service agreement of this type when a budget is cancelled. (The Adjustment Type (Synch Current) field is on the main page.)

- **FASTPATH:**
  Refer to Budget Billing for more information about budgets in general. Refer to Budget Billing Segmentation and Designing and Defining Budget Plans for more information.

Set the **Eligible for Non-Billed Budget** flag to Eligible for Non-Billed Budget if you want SAs of this type to be eligible to be covered by a Non-Billed Budget. If this flag is set to Eligible for Non-Billed Budget, you must also define the **Adjustment Type (Current = 0)** field (on SA Type - Detail).

- **FASTPATH:**
  Refer to Current Amount For SAs Covered By A Non-Billed Budget and SA Types for SAs Covered by Non-Billed Budgets for more information.

Require Total Amount Switch versus Bill Segment Algorithm

The following table shows valid combinations of the SA type's required total amount switch and the bill segment creation algorithm defined on the SA type's bill segment type. If N/A appears in a cell, the combination is not supported in the system. Otherwise, we list typical types of service agreements that will use a combination.

Bill Segment Create Algorithm	**SA Type Require Total Amount Switch:** Not allowed	**SA Type Require Total Amount Switch:** Required
Apply Rate To Usage Request	Metered services. Lamp services. Misc item services. Company usage. Misc recurring charges whose value is specified in a rate or is taxable.	N/A
Recurring Charge With Auto Stop	Payment arrangements. Merchandise installment plans. Zero-interest loans.	N/A
Recurring Charge For Amount To Bill	N/A	Deposit installment plans.

Recurring Charge	Charitable contributions.	N/A
Billable Charge	One time invoices. Pass through charges	N/A
Loan	N/A	Loans.

Allow Recurring Charge Switch versus Bill Segment Algorithm

The following table shows valid combinations of the SA type's allow recurring charge switch and the bill segment creation algorithm defined on the SA type's bill segment type. If N/A appears in a cell, the combination is not supported in the system. Otherwise, we list typical types of service agreements that will use a combination.

Bill Segment Create Algorithm	**SA Type Recurring Charge Switch:** Not allowed	**SA Type Recurring Charge Switch:** Optional	**SA Type Recurring Charge Switch:** Required
Apply Rate To Usage Request	Metered services - no budget Lamp services Misc item services. Company usage. Misc recurring charges whose value is specified in a rate or is taxable.	Metered services - budget optional	Metered services - budget required
Recurring Charge With Auto Stop	N/A	Payment arrangements. Merchandise installment plans. Zero-interest loans. SEE NOTE!	Payment arrangements. Merchandise installment plans. Zero-interest loans. SEE NOTE!
Recurring Charge For Amount To Bill	N/A	Deposit installment plans. SEE NOTE!	Deposit installment plans. SEE NOTE!
Recurring Charge	Charitable contributions	N/A	N/A
Billable Charge	One time invoices. Pass through charges	N/A	N/A
Loan	N/A	N/A	Loans.

➤ **NOTE:**
Most recurring charge SA types require a recurring charge amount on their service agreements. However, the above matrix indicates you can have recurring charge SA types where this value is optional. Why? A special algorithm exists in billing that says if the recurring charge amount is 0 (zero) the system will bill the remaining payoff balance or total amount to bill. This algorithm exists so that you can easily bill the amount in one lump sum (i.e., don't bill it in installments).

Where Used

The billing information is used when the system creates a bill segment for service agreements of this type.

SA Type - Rate

Open **Admin** > **Customer** > **SA Type** > **Search** and navigate to the **Rate** page to define the rates that may be referenced on service agreements of a given type.

Description of Page

Turn on **Rate Required** if the bill segment creation algorithm for the SA type expects a rate schedule to be referenced on service agreements of this type.

> **FASTPATH:**
> For more information, refer to **Rates**.

Define the date the system uses when selecting an effective-dated rate (from the service agreement's rate history) using **Rate Selection Date**. Selecting Bill Start Date will cause the system to use the rate effective on the first day of the bill segment's consumption period. Selecting Bill End Date will cause the system to use the rate effective on the last day of the bill period. Selecting Accounting Date will cause the system to use the rate effective on the accounting date associated with the bill.

If the Contract Management module is not turned off, which includes umbrella agreement management, indicate the **Rate Source**. Check SA Only indicates that the rate schedule currently in effect for the service agreement should always be used. Check TOS First, then SA indicates that if the service agreement is linked to a terms of service record, the service agreement's rate could be overridden by one linked to the terms of service record.

The information in the **Rate Schedules** collection defines the rates that may be referenced on service agreements of this type. The following fields are required for each SA Type:

Rate Schedule Specify the rate schedule; its description is displayed adjacent.

Use As Default Turn on this switch for the rate to be defaulted on new service agreements.

Rate Required versus Bill Segment Algorithm

The following table shows appropriate combinations of the SA type's rate required switch and the bill segment creation algorithm defined on the SA type's bill segment type. If N/A appears in a cell, the combination is not applicable. Otherwise, we list typical types of service agreements that will use a combination. Be aware that no cross validation exists between the rate required switch and the bill segment creation algorithm when setting up the SA type.

Bill Segment Create Algorithm	**SA Type Rate Required Switch:** Not allowed	**SA Type Rate Required Switch:** Rate required on SA
Apply Rate To Usage Request	N/A	Metered services. Lamp services. Misc item services. Company usage.
Recurring Charge With Auto Stop	Payment arrangements. Merchandise installment plans. Zero-interest loans.	N/A

Recurring Charge For Amount To Bill	Deposit installment plans.	N/A
Recurring Charge	Charitable contributions.	N/A
Billable Charge	One time invoices.	Billable charges that require a rate to add-on extra charges (like taxes) or billable charges where the consumption is interfaced and the system is responsible for calculating the charges.
Loan	Loans.	N/A

Where Used

This information is used to default and validate the rate specified on a service agreement. Refer to Service Agreement - Rate Info for more information.

SA Type - SP Type

Open **Admin** > **Customer** > **SA Type** > **Search** and navigate to the **SP Type** page to define the service point types that may be referenced on service agreements of a given type.

Description of Page

Turn on **Service Points Required** if at least one service point should be linked to service agreements of this type in order to properly bill the service agreements.

The information in the **SP Types** collection defines the service point (SP) types that may be referenced on this SA type's service agreements. The following fields are required for each SA Type:

SP Type Specify the SP type; its description is displayed adjacent.

- **FASTPATH:**
  For more information about SP types, refer to Designing SP Types.

Where Used

This information is used to validate the types of service points linked to a service agreement. Refer to Service Agreement - SA / SP for more information.

SA Type - Adjustment Profiles

Open **Admin** > **Customer** > **SA Type** > **Search** and navigate to the **Adj Profile** page to define the adjustment profiles that define adjustment types that may be referenced on service agreements of a given type.

Description of Page

Define the **Adjustment Type Profiles** that, in turn, define adjustment types that may be referenced on service agreements of a given type.

- **FASTPATH:**

For more information about adjustment type profiles, refer to Setting Up Adjustment Type Profiles.

Where Used

This information is used to validate the adjustments linked to the service agreement. Refer to Adjustments - Main Information for more information.

SA Type - C&C

Open **Admin** > **Customer** > **SA Type** > **Search** and navigate to the **C&C** page to maintain attributes that affect how the system severs the service agreement when collection attempts fail.

FASTPATH:
Refer to Designing Your Severance Procedures for more information.

Description of Page

Select the **Debt Class** associated with the SA Type. Any debt on a service agreement of this SA Type will be categorized under this debt class.

Select the **Write Off Debt ClassCode** associated with the SA Type. Any debt on a service agreement of this SA Type will be categorized under this debt class during write-off processing.

NOTE:
Write Off Debt Class vs. Regular Debt Class. It's important to be aware that a SA type references both a regular debt class and a write-off debt class. The regular debt class controls the collection criteria applied against an account's service agreements. The regular debt class is also used to segregate an account's outstanding balance on several queries in the system. The write-off debt class controls the write-off criteria applied against an account's stopped service agreements. The reason the system supports two different debt classes is because you may categorize your service agreements differently when you try to collect overdue debt versus when you write-off debt. Refer to The Big Picture Of Write Off Processing for more information.

The information in the **Severance Criteria** collection defines the SA Type's severance criteria. Severance criteria define the severance process to be executed for service agreements of a given SA type. The severance process may differ depending on some attribute of the customer or premise. For example, you may have a different severance process if the customer has life support equipment.

FASTPATH:
The following information is not intuitively obvious. Refer to Designing Your Severance Procedures for more information.

The following fields are required for each instance:

Priority The priority controls the order in which the system determines if the severance process should be applied (the first severance process whose algorithm applies is used). Higher priorities are checked before lower priorities.

Severance Criteria Algorithm Select the algorithm to be used to check if the severance process should be initiated for service agreements of this type. If a condition is met, a severance process is created using the associated severance process template.

If you haven't done so already, you must set up this algorithm in the system. To do this:

- Create a new algorithm (refer to Setting Up Algorithms).
- On this algorithm, reference an Algorithm Type that determines if the severance of a service agreement should be processed using the associated **Severance Process Template**. Click here to see the algorithm types available for this plug-in spot.

△

IMPORTANT:
You must have at least one entry in this collection otherwise the system will not start a severance process when a service agreement of this type needs to be stopped due to non payment. This entry should have the lowest priority code and should reference a **Severance Criteria Algorithm** that references the SV CRIT DFLT algorithm type.

Severance Process Template Specify the severance process template to use to sever the service agreement; its description is displayed adjacent.

Where Used

The debt class has multiple uses:

- The system summarizes an account's debt by debt class on Account - Main Information and Account - Financial Balances.
- Debt class is one component that controls how the system analyzes an account's overdue debt (the others are the account's collection class and currency). Refer to Different Collection Criteria For Different Customers And Different Debt for more information.
- Write off debt class is one component that control how the system writes off an account's stopped service agreements. Refer to Different Write-Off Criteria For Different Customers And Different Debt for more information.

The severance criteria are used when a collection event is activated that indicates that service should be severed.

SA Type - Billable Charge Template

Open **Admin** > **Customer** > **SA Type** > **Search** and navigate to the **BC Template** page to define the billable charge templates that can be used on service agreements of a given type.

➤

NOTE:
Only billable charges have billable charge templates. Only service agreements that are defined as Billable Charges (in the Special Role on the Details window) may use the grid on this window.

Description of Page

The information in the **Billable Charge Template** collection defines the SA Type's permissible billable charge templates. A billable charge template contains the default bill lines, amounts and distribution codes used to levy a one-off charge. The following fields are required for each template:

Billable Charge Template Specify the billable charge template. Its description is displayed adjacent.

Use As Default Turn on this switch for the template to be defaulted on new billable charges linked to service agreements of this type (if any).

● **FASTPATH:**
For more information about billable charge templates, refer to Setting Up Billable Charge Templates.

Where Used

This information is used to limit the billable charge templates that can be used for a given SA type.

SA Type - Characteristics

To define characteristics common to all service agreements of a given type, open **Admin** > **Customer** > **SA Type** > **Search** and navigate to the **Characteristics** page.

Description of Page

Use the characteristics collection to define a **Characteristic Type**, **Sequence**, and **Characteristic Value** common to all service agreements of this type.

➤ **NOTE:**
You can only choose characteristic types defined as permissible on a SA Type record. Refer to Setting Up Characteristic Types & Their Values for more information.

SA Type - Algorithms

Open **Admin** > **Customer** > **SA Type** > **Search** and navigate to the **Algorithm** page to define the algorithms that should be executed for service agreements of a given type.

Description of Page

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 (descriptions of all possible events are provided below).
- 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.

WARNING:
These algorithms are typically significant processes. The absence of an algorithm may prevent the system from operating correctly.

The following table describes each **System Event** for which you can define algorithms.

System Event	Optional / Required	Description
Bill Completion	Optional	These algorithms are executed whenever a bill is completed for an account that contains

		a non-canceled service agreement of this type. The following situations necessitate the definition of a completion algorithm on an SA type: - As explained under Technical Implementation of A/R Transfer and Technical Implementation of Routing Billable Charges To Service Providers, when a bill is completed, the system needs to set up the data necessary to interface any "master" SA's charges to the service provider and to transfer the receivable balance from the customer to the service provider. The system will only do this if you specify an appropriate algorithm on the master SA types. - As explained under Billing For SAs Covered By The Non-Billed Budget, when a bill is completed for accounts that have a Non-Billed Budget SA, the system needs to distribute the Non-Billed Budget's credit balance to the covered SAs. The system will only do this if you specify an appropriate algorithm on the NBB SA types. - As explained under Overpayment Segmentation, when a bill is completed, the system may apply an excess credit from a prior overpayment to an account's service agreements. **Note**. Algorithms of this type are called for all non- Canceled service agreements, regardless of whether or not they are billed. If your algorithms should only be processed under certain conditions (for example, only process this algorithm for Active service agreements), then it is the responsibility of the algorithm to check the conditions before continuing. Click here to see the algorithm types available for this system event.
Break NBB SA	Optional	These algorithms are executed when a Non-Billed Budget is manually stopped via the **Non-Billed Budget maintenance page**. Click here to see the algorithm types available for this system event. Note that the Payment Arrangement algorithm and the Break Pay Arrangement algorithm are mutually exclusive.
Break Pay Arrangement	Optional	These algorithms are by executed by severance events when the event is created for payment arrangement SAs. This algorithm should be specified on SA types with a

		special role of Payment Arrangement to perform special actions that take place when a customer breaks a payment arrangement. Refer to Monitoring Payment Arrangements for more information about breaking payment arrangements. Click here to see the algorithm types available for this system event.
Budget Eligibility	Optional	These algorithms are executed when determining in a service agreement is eligible for budget. Algorithms of this type are only applicable on SA types that are marked as eligible for budget and may be used to override that setting and indicate that the service agreement is not eligible. For example, maybe service agreements in a certain rate are not eligible. Or perhaps service agreements with a given characteristic value are not eligible. Click here to see the algorithm types available for this system event.
Cut Process Rule	Optional	These algorithms are executed to create a cut process for service agreements of this type. Refer to The Big Picture Of Cut Processes for more information. Click here to see the algorithm types available for this system event.
FT Freeze	Optional	These algorithms are executed whenever a financial transaction is frozen that is linked to a service agreement of this type. The following situations necessitate the definition of an FT freeze algorithm: - As explained under Technical Implementation of Routing Consumption To Service Providers, when a master SA's bill segment is frozen, the system must check if there are any service providers who need the bill segment's consumption. If so, it sets up the data necessary to interface the master SA's consumption (snapshot on the bill segment) to the service provider(s). The system will only do this if you specify an appropriate FT Freeze Algorithm on the master SA types. - As explained under Technical Implementation of Paying The Service Provider, when a financial transaction (FT) is frozen that is associated with a sub SA, the system must check if this FT should trigger the "payment" of a service provider. If so, it has to create an adjustment to increase

		how much we owe the service provider. The system will only do this if you specify an appropriate FT Freeze Algorithm on the sub SA types. Click here to see the algorithm types available for this system event.
High Bill Amount	Optional	This algorithm type checks for high bill amounts during batch billing. A bill threshold amount can be defined on a service agreement. By default the system compares this threshold amount to the bill segment amount during batch billing. If the bill segment amount exceeds the given threshold value, a bill error is generated. No proration takes place on the threshold amount if the bill segment's consumption period falls outside of the rate frequency's normal period. This algorithm may be used to override the default system behavior. It will prorate the SA's bill threshold amount before comparing it to the batch bill segment amount if the consumption period falls outside of the rate frequency's normal period. Click here to see the algorithm types available for this system event.
Landlord Reversion	Optional	These algorithms are used to create, update or cancel a service agreement for a landlord when service is started, stopped or updated for a premise that references a landlord agreement. Algorithms of this type are called: - When starting a new SA (start initiation) - When stopping an SA (stop initiation) - When canceling a pending start SA - When changing the stop / start dates of back-to-back service agreements Click here to see the algorithm types available for this system event.
Loan Interest Charge	Optional	These algorithms are executed whenever the interest charge needs to be calculated for a loan, such as when the loan amortization schedule is created and when a bill segment is created for a loan SA. This algorithm should be specified on SA types with a special role of Loan. Refer to Defining Loan Options for more information. Click here to see the algorithm types available for this system event.
Loan Periods and Amount	Optional	These algorithms are executed whenever a user clicks the Calculate button on **Loan -**

		Main or on the Start SA Confirmation dialog for a loan SA. It calculates either the number of periodic payments or the payment amount (depending on whether the user specifies the number of payments or the payment amount as input). This algorithm should be specified on SA types with a special role of Loan. Refer to Defining Loan Options for more information. Click here to see the algorithm types available for this system event.
Loan Schedule	Optional	These algorithms are executed whenever the system needs to create a **loan amortization schedule** for a loan, such as when you renegotiate the terms of a loan on **Loan - Main**. This algorithm should be specified on SA types with a special role of Loan. Refer to Defining Loan Options for more information. Click here to see the algorithm types available for this system event.
Payment Arrangement	Optional	These algorithms are by executed to handle the creation, breaking and canceling of payment arrangement SAs. This algorithm should be specified on SA types with a special role of Payment Arrangement to perform special actions that take place during the lifecycle of a payment arrangement. Refer to Monitoring Payment Arrangements for more information about payment arrangements. Note that the Payment Arrangement algorithm and the Break Pay Arrangement algorithm are mutually exclusive.
Payment Freeze	Optional	These algorithms are executed whenever a payment is frozen. The following situations necessitate the definition of such an algorithm: - For a loan SA, such an algorithm is required to create a frozen adjustment that transfers any credit balance resulting from an overpayment to the loan's principal balance. Refer to Defining Loan Options for more information. Click here to see the algorithm types available for this system event.
Pre-Bill Completion	Optional	These algorithms are executed immediately prior to bill completion when a bill contains a bill segment for a service agreement whose SA type has such an algorithm. The following situations necessitate the definition of such an algorithm: - If you want to delete a bill segment that's in error on the last night of a bill cycle when

		there are other bill segments that aren't in error, use such an algorithm. Click here to see the algorithm types available for this system event.
Process NBB Scheduled Payment	Optional	These algorithms are executed by the NBB Scheduled Payment Processing background process whenever a scheduled payment is due. If the Non-Billed Budget SA is unmonitored, this algorithm is not called. This algorithm should be specified on Non-Billed Budget SA types to create the necessary adjustments for the Non-Billed Budget SA. Click here to see the algorithm types available for this system event.
Proposal SA Acceptance	Optional	These algorithms are executed when a proposal service agreement is accepted. Refer to Enabling The Creation Of A Real Service Agreement for more information. Click here to see the algorithm types available for this system event.
Proposal SA Bill Segment Generation	Optional	These algorithms are executed to generate simulated bills segments for a **proposal service agreement**. Refer to **Enabling The Generation Of Simulated Bill Segments** for more information. Click here to see the algorithm types available for this system event.
Proposal SA Creation	Optional	These algorithms are executed when a **proposal service agreement** is created. Refer to **Enabling The Automatic Generation Of Billing Scenarios** for more information. Click here to see the algorithm types available for this system event.
SA Activation	Optional	These algorithms are executed when a service agreement status changes from Pending Start to Active. It performs any additional activities that are necessary to activate an SA. The following situations necessitate the definition of such an algorithm: - If you want to create a customer contact to indicate that a non-billed budget has been activated, use such an algorithm. Click here to see the algorithm types available for this system event.
SA Cancel	Optional	These algorithms are executed when a service agreement status changes to Canceled. It performs any additional activities that are necessary to cancel an SA.

		An example of when you may use this algorithm is that perhaps your business rules dictate that the creation of a payment arrangement should create a credit rating history transaction. When a payment arrangement SA is canceled, the credit rating should be updated with an end date. Click here to see the algorithm types available for this system event.
SA Creation	Optional	These algorithms are executed when a service agreement is created. The following situations necessitate the definition of such an algorithm on an SA type: - If you want to create a To Do entry whenever a new service agreement of a given type is added, specify such an algorithm. - If you want to automatically activate SAs of a given type (instead of waiting for the background SA activation process to run), specify such an algorithm. - If you want to create a Workflow Process when a service agreement of a given type is added, specify such an algorithm. Click here to see the algorithm types available for this system event.
SA Information	Optional	We use the term "SA information" to describe the basic information that appears throughout the system to describe a service agreement. The data that appears in "SA information" is constructed using this algorithm. Plug an algorithm into this spot to override the "SA information" algorithm on installation options or the system default "SA information" if no such algorithm is defined on installation options. Click here to see the algorithm types available for this system event.
SA Renewal	Optional	These algorithms are executed by the Service Agreement Renewal background process whenever an SA is due for renewal or when the user clicks the **Renew** button (for **Non-Billed Budgets**). It performs any activities that are necessary to renew an SA and returns the new renewal and expiration dates for the SA. Click here to see the algorithm types available for this system event.
SA Stop	Optional	These algorithms are executed whenever a service agreement's status changes from pending stop to stopped. The following

		situations necessitate the definition of a SA stop algorithm: - For Non-Billed Budgets to distribute any remaining credit balance from the Non-Billed Budget SA to the covered SAs, you must specify such an algorithm. - For service credit memberships that have a refundable membership fee, an SA Stop algorithm attempts to refund the fee if this is the last SA linked to the membership that is being stopped. Click here to see the algorithm types available for this system event.
SA Stop Initiation	Optional	These algorithms are executed whenever a service agreement's status becomes pending stop. The following situations necessitate the definition of a stop initiation algorithm on an SA type: - As explained under Finalizing Pending Stops, service agreements are normally transitioned from pending stop to stopped by a background process (or manually). For Non-Billed Budget SAs to transition to stopped automatically (without waiting for the background process), you must specify such an algorithm. **When does a SA become pending stop?** Service agreements typically become pending stop when a user initiates a request to stop service on Start Stop - Main. A severance process with an Expire SA severance event causes a service agreement to become pending stop (when the event is executed). Additionally, the Stop Expired Service Agreements background process starts the process to initiate the stop of an SA when the expiration date is on or before the process date. Click here to see the algorithm types available for this system event.
Start Stop Field Work	Optional	These algorithms are executed to create the field activities necessary to start and stop service. Refer to Starting Service and Field Activities and to Stopping Service and Field Activities for a description of when algorithms of this type are called. The following situations necessitate the definition of a start stop fieldwork creation algorithm: - If a service agreement has field activities created to start and stop service at its service

points, its SA type must have an appropriate start stop field work creation algorithm.

Click here to see the algorithm types available for this system event.

SA Type - Billable Charge Overrides

The BCU2 - Create Billable Charge background process is responsible for creating billable charges for each billable charge upload staging record interfaced into the system. This process will override the values of the various switches referenced on a billable charge upload staging line if the respective service agreement's SA type has an override value for the billable charge upload staging line's billable charge line type.

> **NOTE:**
> **This information is optional.** If you don't need to override the values of a **Billable Charge Line Type** you don't need to set up this information.

Open **Admin** > **Customer** > **SA Type** > **Search** and navigate to the **BC Upload Override** page to define override values for a given SA Type / Billable Charge Upload Staging Line Type.

Description of Page

Use the **Billable Charge Overrides** collection to define values to be overridden on billable charge lines uploaded from an external system (refer to the description above for the details). The following switches may be overridden for a given **SA Type** and **Billable Charge Line Type.**

- Use the **Show on Bill** switch to define the value to be defaulted into the Show on Bill indicator on billable charge upload lines that reference this line type.
- Use the **Appear in Summary** switch to define the value to be defaulted into the App in Summary indicator on billable charge upload lines that reference this line type.
- Use **Memo Only, No GL** switch to define the value to be defaulted into the Memo Only, No GL indicator on billable charge upload lines that reference this line type.
- Use **Distribution Code** to define the value to be defaulted into the Distribution Code on billable charge upload lines that reference this line type.

SA Type - Contract Option Type

Open **Admin** > **Customer** > **SA Type** > **Search** and navigate to the **Contract Option Type** page to define the contract option types, which are valid for service agreements of a given type.

> **NOTE:**
> **This tab may not appear.** This tab is suppressed if the interval billing Contract Options module is turned off.

Description of Page

If the SA Type’s special role flag is set to Bill Determinants Required, you may define the **Contract Option Types** that are valid for contract options linked to a service agreement of this type.

Where Used

This information is used to validate the contract options linked to the service agreement. Refer to Service Agreement - Contract Options for more information.

SA Type - NBB Recommendation Rule

Open **Admin** > **Customer** > **SA Type** > **Search** and navigate to the **NBB Rec'n Rule** page to define the recommendation rules that are valid for Non-Billed Budget SAs of this type.

Description of Page

If the SA Type's special role is Non-Billed Budget, you may define the **Recommendation Rules** that are valid on non-billed budget SAs of this type. Check the **Use As Default** box to indicate the default recommendation rule for service agreements of this type.

●

FASTPATH:
For more information about Non-Billed Budgets, refer to **Defining Non-Billed Budget Options**.

Where Used

The Non-Billed Budget recommendation rules are used to recommend the payment amount and payment schedule for Non-Billed Budget service agreements. Refer to **Maintaining Non-Billed Budgets** for more information.

Setting Up Start Options

Start options save users time and prevent data entry errors because they default many values on a service agreement (e.g., the rate schedule, recurring charge amount, contract riders, contract terms, characteristics, terms and conditions, etc. can all be defaulted onto a service agreement from a start option).

A SA type may have zero or more start options.

- A SA type without start options is usually one that has a very limited number of options. For example, if a SA type has a single valid rate and no customer-specific contract values, you don't need to setup a start option (the SA's default rate can default based on the information defined when you setup the SA type).
- A SA type with multiple start options is one where many different permutations are possible. For example, a SA type that can have multiple rates and each rate can have multiple riders is a good candidate for start options (where each start option will default, for example, a specific rate and set of contract riders).

When the Start/Stop transaction is used to start service AND the service being started uses a SA type with start options, the user is asked to select one of the start options. The service agreement that's created is populated with fields from the start option.

When the Order transaction is used to start service, the user selects start options, but only indirectly. It works like this:

- A user selects a "package" of services to start service for a customer.
- A "package" contains one or more start options.
- When a user selects a package, the system creates a service agreement for each start option on the package (and defaults the information on each service agreement from the respective start option).

A start option's default values may change over time (i.e., the information on a start option is effective-dated). The start service logic uses the version of the start option that is effective on the day service starts.

Start options can cause a great deal of information to be populated on a service agreement. There are several ways to change this default information:

- A user may override this information using the Service Agreement transaction.
- If the service agreement is in the pending start state, you can use the Start/Stop - Pending SAs page to change the service agreement's SA type and/or start option.
- If the service agreement is in active or pending stop states, a button appears on Service Agreement - Main called **Apply New Start Option**. When pressed, the user is allowed to define a start option and the date its terms become effective on the service agreement. Refer to Changing A Start Option for the details of this functionality.

The topics in this section describe how to setup start options.

> **NOTE:**
> **The merge transaction can save setup time.** The Start Options Merge transaction can be used to construct a start option by copying pieces from other start options.

Start Option Considerations - Rate-Oriented SA Types

To understand the following discussion, you should be familiar with the following concepts:

- SA types that use rates have one or more valid rate schedules. Only these rate schedules may be defined on service agreements of a given type.
- One of a SA type's rate schedules may be designated as the "default" rate. The system assigns the "default" rate to new service agreements when a CSR doesn't choose a start option.
- Start options may be used by customer service reps when a service is started for a customer. A start option causes the customer's new service agreement to be populated with a specific rate and contract terms (e.g., contract riders, contract values). The use of start options is not allowed if the service's SA type does not require a start option. The use of start options is required if the service's SA type requires a start option.

Whether or not a SA Type has start options is dependent on the following factors:

- If a SA Type has only one valid rate and the rate doesn't use customer-specific contract terms (e.g., contract riders, contract values), the SA Type does not need any start options. Why? Because the system default's the SA type's default rate on new service agreement when no start option is used at start time.
- If a SA Type has only one valid rate, but under unusual circumstances, it uses customer-specific contract terms, you'll want an option for every situation (both the standard one and the unusual ones).
- If a SA Type has a single rate with a variety of contract terms (which, by definition, are different for each customer), then you'd want a start option for each permissible combination of contract terms. You'd also want to turn on the SA type's Start Option Required switch to make your CSR's pick one of the start options when service is started (rather than let the system use the SA type's default rate).
- If a SA type has multiple valid rates and a variety of contract terms (a combination of the previous two points), you'd need a start option for each permissible combination. You'd also want to turn on the SA type's Start Option Required switch to make your CSR's pick one of the start options when service is started (rather than let the system use the SA type's default rate).

Start Option Considerations - SA Types That Use Recurring Charges

To understand the following discussion, you should be familiar with the following concepts:

- Many SA types use the recurring charge algorithms to generate the bill segments. For example, if you let a customer pay for a $900 heat pump in 3 installments of $300, you'd have an SA type called CA/MERCH-I and indicate it uses the recurring charge algorithm.
- When a CSR creates an SA type that uses recurring charge algorithms, they can enter the recurring charge amounts on Start/Stop Service - Start Confirmation window OR they can specify a start option on this window and let the system populate the recurring charge amount. For example, if you let a customer pay for a $900 heat pump in 3 installments of $300, you can set up a start option called HP 3PAY with an installment amount of $300.
- Depending on the type of service being started, you might also need to generate an adjustment when service is started in order to initialize the total debt. For example, when a customer buys a heat pump we'll need an adjustment issued to realize the entire $900 of revenue.

The following table provides examples of recurring charge SA types with several typical start options:

CIS Division/ SA Type	Start Option	Adjustment Type	Install Amount	Comments
CA/CHARITY	DONATE $5		5	This causes a charitable contribution to be created with an installment amount of $5.
CA/CHARITY	DONATE $10		10	This causes a charitable contribution to be created with an installment amount of $10.
CA/PA-REGU	PAY $10 PM		10	This causes a payment arrangement to be created with an installment amount of $10.
CA/PA-REGU	PAY $20 PM		20	This causes a payment arrangement to be created with an installment amount of $20.
CA/PA-UNRE	PAY $10 PM		10	This causes a payment arrangement to be created with an installment amount of $10.
CA/PA-UNRE	PAY $20 PM		20	This causes a payment arrangement to be created with an installment amount of $20.
CA/MERCH-I	HP 3PAY	HEATPUMP	300	This causes a merchandise service

				agreement to be created with an installment amount of $300. It also causes an adjustment to be issued to realize the $900 of revenue.
CA/MERCH-I	BBQ 3PAY	BBQ	250	This causes a merchandise service agreement to be created with an installment amount of $250. It also causes an adjustment to be issued to realize the $750 of revenue.
CA/DEP-I	PAY $40 PM		40	This causes a deposit service agreement to be created with an installment amount of $40. The CSR would be required to define the total deposit amount to be billed over the life of the service agreement on the secondary SA window on the Start Account window.

Start Option Considerations - Initial Adjustment SA Types

To understand the following discussion, you should be familiar with the following concepts:

- Some SA types depend on an adjustment to book their initial debt. For example, the CA/MERCH-I SA type requires an initial adjustment to book the payoff amount for the entire cost of the merchandise. Another example would be a loan service agreement (refer to Booking The Principal Amount Using An Adjustment for the details).
- When a CSR starts an SA type that requires an initial adjustment, they can create the adjustment immediately after starting service OR then can specify a start option when they start service and let the system generate the adjustment. You could let the system create the adjustment to book the $900 associated with the heat pump as shown in the previous section.

The following table provides examples of SA types that use adjustments with several typical start options:

CIS Division/ SA Type	Start Option	Adjustment Type	Install Amount	Comments
CA/MERCH-I	HP 3PAY	HEATPUMP	300	This causes a merchandise service agreement to be created with an installment amount of $300. It also causes an adjustment to be issued to realize the $900 of revenue.

CA/MERCH-I	BBQ 3PAY	BBQ	250	This causes a merchandise service agreement to be created with an installment amount of $250. It also causes an adjustment to be issued to realize the $750 of revenue.
CA/CONNECT	CONNECT	CONNECT		This causes an adjustment to be issued to realize the connection charge.

Start Option - Main

Open **Admin** > **Customer** > **SA Type Start Option** > **Add** to define a SA type's start options.

Description of Page

Every start option is uniquely identified by the following fields:

CIS Division & SA Type Enter the Division and SA type to which the start option is linked.

Start Option Enter the unique identifier of the option. Pick something easy to recognize as this will be used by CSRs to pick an option when they start service.

Effective Date Enter the earliest effective date. It should be the same as the earliest effective date of the start option's rate (although it doesn't hurt for it to be earlier). The date defaults to the current date. (The status, below, should be Active.)

The remaining fields further describe a start option.

Enter a **Description** for the start option.

Indicate its **Status**. For new start options, the status should be Active. When it's no longer applicable, change it to Inactive.

Enter the primary **Rate Schedule** that should be defaulted onto service agreements created using this option. Refer to Start Option Considerations For SA Types That Use Rates for more information.

FASTPATH:
For more information about a service agreement's rates, refer to Service Agreement - Rate Info.

Enter the **Adjustment Type** that should be generated, if any, when service is started using this option. Refer to Start Option Considerations For SA Types That Use Initial Adjustments for more information.

Enter the **Recurring Charge Amount** that should be defaulted onto service agreements created using this option. This field is only visible when the SA type allows recurring charges. In addition, the prompt for this field is defined on the SA type table on the billing window (e.g., it could appear as Payment Amount, Budget Amount, or Installment Amount). Refer to Start Option Considerations For SA Types That Use Recurring Charges for more information.

Enter the **Currency Code** in which monetary amounts are denominated.

NOTE:

The currency code defaults from the installation record.

Enter the **Total Amount to Bill** that should be defaulted onto service agreements created using this option. This is useful to initiate either a loan or a deposit. The prompt for this field is defined on the SA type table on the billing window (e.g., it could be Deposit Amount or Loan Amount).

Use **Number of Periods** to default the number of payment periods of service agreements created using this option. This field is only allowed for SA types with special role of Loan. Refer to The Terms Of A Loan Are Stored On A Service Agreement for more information.

If your SA Type has a special role of Billing Determinants Required, you may enter the **Cutoff Time** and **Start Day Option** that should be defaulted onto service agreements that are created using this option. For start day option, you may choose **Current Day** or **Previous Day**.

If your SA Type has a special role of Billable Charge, then you can setup the start option to automatically create a billable charge when a service agreement is created using this start option. For example, you might have a start option that automatically creates a "one-time invoice" service agreement along with a "tree trimming" billable charge. To use this feature you should turn on the **Create Billable Charge** switch and specify the **Billable Charge Template** that will be used to create the billable charge. These fields are only allowed for SA types with special role of Billable Charge.

- **FASTPATH:**
  Refer to Setting Up Billable Charge Templates for more information about templates. Refer to An Easier Way To Create One Time Charges for an example of how you can setup a campaign with packages that use this type of start option.

➤ **NOTE:**
The duplicate action in the page actions toolbar enables you to copy another start option. Refer to Duplicate Button in the system wide standards document for more information.

Start Option - Rate Info

Open **Admin** > **Customer** > **SA Type Start Option** > **Search** and navigate to the **Rate Info** page to define the start option's default values for contract riders and contract values.

Description of Page

The information in the **Contract Riders** collection defines the contract riders to be defaulted onto service agreements created using this start option. The following fields are required for each instance:

Bill Factor The bill factor defines the type of rider. You may only reference bill factors designated as being applicable for contract riders.

Number of Days The number of days the rider should be in effect. This value is used by the system to set the stop date on the service agreement's contract rider. If the rider has no expiration, set this field to 0. Default note: this field will be set to 0 if left blank.

- **FASTPATH:**

For more information about a service agreement's contract riders, refer to Service Agreement - Contract Riders.

The information in the **Contract Values** collection defines the contract values to be defaulted onto service agreements created using this start option. The following fields are required for each event:

Bill Factor The bill factor defines the type of value. You may only reference bill factors designated as allowing values in contract terms.

Number of Days The number of days the value should be in effect. This value is used by the system to set the stop date on the service agreement's contract value. If the value has no expiration, set this field to 0.

Value The amount of the contract value.

- **FASTPATH:**
  For more information about a service agreement's contract values, refer to Service Agreement - Contract Values.

Start Option - Characteristics & Qty

Open **Admin** > **Customer** > **SA Type Start Option** > **Search** and navigate to the **Characteristics & Qty** page to define the start option's default values for characteristics and contract quantities.

Description of Page

The information in the **Characteristics** collection defines the characteristics to be defaulted onto service agreements created using this start option. The following fields are required for each instance:

Characteristic Type This defines the type of characteristic. Note: you may only define characteristics valid on service agreements.

Characteristic Value This defines the characteristic value that will be defaulted.

- **FASTPATH:**
  For more information about a service agreement's characteristics, refer to Service Agreement - Characteristics.

The information in the **Contract Quantity** collection defines the contract quantities to be defaulted onto service agreements created using this start option. The following fields are required for each instance:

Contract Quantity Type This defines the type of contract quantity.

Contract Quantity The amount of the contract quantity.

- **FASTPATH:**
  For more information about a service agreement's contract quantities, refer to Service Agreement - Contract Quantity.

Start Option - Contract Option

> **NOTE:**
> **This tab may not appear.** This tab is suppressed if the interval billing Contract Options module is turned off.

Open **Admin** > **Customer** > **SA Type Start Option** > **Search** and navigate to the **Contract Option** page to define the start option's contract option default values.

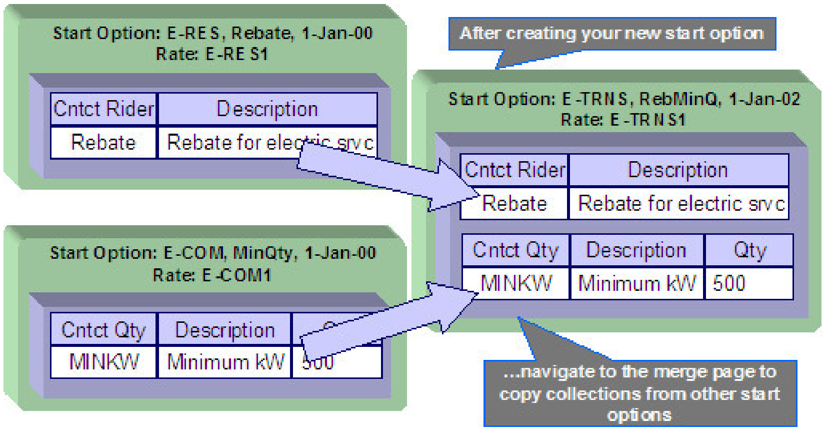
Description of Page

The collection of contract option types will be used to link contract options to a service agreement that will be created with this start option. If the contract option to create for the service agreement will be SA Specific, simply indicate the **Contract Option Type**. This will cause a new contract option to be created with this contract option type and the new contract option will be linked to the new SA at start time. If the contract option to create for the service agreement will be Shared, indicate the contract option type and the **Contract Option ID**. This will cause the specified contract option to be linked to the new SA at start time.

Start Option Merge

Use this page to modify an existing start option by copying information from other start options. This page may be used to copy records from the contract rider, contract value, contract quantity, characteristic, contract option and TOU contract value collections from one or more existing start options to another.

> **NOTE:**
> The target start option must exist prior to using this page. If you are creating a new start option, you must first go to the Start Option page to add the new start option and then navigate to the merge page to copy collection information.



Start Option: E-RES, Rebate, 1-Jan-00
Rate: E-RE S1

Cntct Rider	Description
Rebate	Rebate for electric srvc

Start Option: E-COM, MinQty, 1-Jan-00
Rate: E-COM1

Cntct Qty	Description	Qty
MINKW	Minimum kW	500

After creating your new start option

Start Option: E-TRNS, RebMinQ, 1-Jan-02
Rate: E-TRNS1

Cntct Rider	Description
Rebate	Rebate for electric srvc

Cntct Qty	Description	Qty
MINKW	Minimum kW	500

...navigate to the merge page to copy collections from other start options

➤ **NOTE:**

Duplicate versus Merge. The **Start Option** page has Duplication capability. You would duplicate a start option if you want to a) create a new start option AND b) populate it with all the information from an existing start option. You would use the start option merge page if you want to build a start option using pieces of one or more start options.

Start Option Merge - Main

Open **Admin** > **Customer** > **SA Type Start Option Merge.**

Description of Page

Select the **Original Start Option**, which is the target for merging the start option collection information.

Select the **Merge From Start Option**, which is your template start option to copy the collections from.

➤ **NOTE:**

You may only copy information from one Merge From start option at a time. If you wish to copy information from more than one start option, select the first Merge From start option, copy the desired records, Save, then select the next Merge From start option.

The left portion of the page will display any existing records in the collections for the original start option. The right portion of the page will display the existing records in the collections for the Merge From start option.

You may use the **Copy All** button to copy all the records in all the collections from the Merge From start option to the Original start option. If you do not choose to copy all, you may copy records individually as described below.

The left portion of the **Contract Riders** collection initially displays existing contract riders linked to the original start option. In the **Merge Type**, you will see the word Original, for any of these records. The **Bill Factor** and **Number of Days** for each contract rider is displayed. In the right portion of the collection, the existing records in the merge from start option are displayed initially.

The left portion of the **Contract Values** collection initially displays existing contract values linked to the original start option. In the **Merge Type**, you will see the word Original, for any of these records. The **Bill Factor**, **Number of Days** and **Value** for each contract value is displayed. In the right portion of the collection, the existing records in the merge from start option are displayed initially.

The topics, which follow, describe how to perform common maintenance tasks:

Removing A Row From A Grid

If you wish to remove a record linked to the Original start option, click the "-" button to the left of the record.

Adding A New Row To A Start Option

You may move any of the records from the Merge From start option to the original start option by selecting the left arrow adjacent to the desired row. Once a record is moved it will disappear from the Merge From information and appear in the Original information with the word Merge in the Merge Type column.

Removing An Uncommitted Row From A Start Option

If you have copied a row across by mistake, you may remove it by clicking on the right arrow adjacent to the appropriate record.

Start Option Merge - Characteristics and Quantities

Open **Admin** > **Customer** > **SA Type Start Option Merge** and navigate to the **Characteristics and Quantities** page to copy rows in the characteristic and contract quantity collections.

Description of Page

The left portion of the **Characteristics** collection initially displays existing characteristics linked to the original start option. In the **Merge Type**, you will see the word Original, for any of these records. The **Characteristic Type** and **Characteristic Value** for each characteristic are displayed. In the right portion of the collection, the existing records in the merge from start option are displayed initially.

The left portion of the **Contract Quantity** collection initially displays existing contract quantities linked to the original start option. In the **Merge Type**, you will see the word Original, for any of these records. The **Contract Quantity Type** and **Contract Quantity** for each contract quantity are displayed. In the right portion of the collection, the existing records in the merge from start option are displayed initially.

●

FASTPATH:
Refer to Start Option Merge - Main for more information about how to perform common maintenance tasks for the grids displayed on this page.

Start Option Merge - Contract Option

Open **Admin** > **Customer** > **SA Type Start Option Merge** and navigate to the **Contract Option** page to copy rows in the contract options collection.

➤

NOTE:
This tab may not appear. This tab is suppressed if the interval billing Contract Options module is turned off.

Description of Page

The left portion of the **Contract Option** collection initially displays existing contract option information linked to the original start option. In the **Merge Type**, you will see the word Original, for any of these records. The **Contract Option Type**, **Contract Option ID** and **Description** of the contract option for each contract option row are displayed. In the right portion of the collection, the existing records in the merge from start option are displayed initially.

- **FASTPATH:**
  Refer to Start Option Merge - Main for more information about how to perform common maintenance tasks for the grids displayed on this page.

Start Option Merge - Terms and Conditions

Open **Admin** > **Customer** > **SA Type Start Option Merge** and navigate to the **Terms and Conditions** page to copy terms and conditions (T&Cs).

Description of Page

The left side of the **Terms and Conditions** grid initially displays the T&Cs linked to the original start option. On the right side, the T&Cs linked to the merge from start option are displayed initially.

- **FASTPATH:**
  Refer to Start Option Merge - Main for a description of how to perform common maintenance tasks for the grids displayed on this page.

Defining SA Relationship Options

We use the term "SA Relationship" to describe functionality that supports the following situations:

- When companies other than your own provide a service to your customers AND you have some type of interaction with these companies. For example, in a deregulated market, customers deal with both distribution and energy supply companies. These companies typically exchange a great deal of information about their joint customers.
- When multiple rates are associated with a service (where each rate corresponds with a sub-category of service). For example, most water companies charge for both water and wastewater service using separate rates for each. While it is possible to set up water and wastewater as separate service agreements, the SA relationship functionality allows you to set up a single "master" service agreement (for the water service) and associate with it a "sub" service agreement (for the wastewater service).
- When a party representing a group of customers negotiates a contract that is applied over and above those of the individual service agreements. For example, the head office of a national chain may negotiate for additional discounts that should be calculated together or individually. The SA relationship functionality may be used to track the covered service agreements and to calculate and transfer discounts to the head office's service agreement. Refer to the special discounts section.

WARNING:
Setting up the SA relationship control tables is as challenging as your organization's business rules. If you don't have requirements similar to those described above, you don't have to set up anything. If you have these types of requirements, your setup process will be taxing as you must design and set up control tables that manage the financial and consumption interactions that take place between you, your customers, and the various service providers.

The topics in this section describe tables that control your SA relationship functionality.

The Big Picture of SA Relationships and Service Providers

You must set up service providers if companies other than your own provide a service to your customers AND you have some type of interaction with these companies. You will have one service provider for each such company. For example, if you are a distribution company in a deregulated market, you will have a service provider for each company that provides any of the following services:

- Energy (commonly referred to as Energy Supply Companies, Energy Service Providers, Retailers, and Suppliers)
- Meter service (commonly referred to as Meter Service Providers and Meter Agents)
- Meter reading (commonly referred to as Meter Data Management Agencies and Meter Reading Service Providers)
- Billing (commonly referred to as Billing Agents)

The topics in this section provide background information about service providers.

Persons and Service Providers

A great deal of information about your service providers is defined using a person. For example, a service provider's name, address, phone numbers, electronic ID's, etc. are all defined on the person object.

In addition, every service provider must have a service provider object created for it. The service provider object contains information about a provider's relationships with the customer and your organization, for example:

- Do you calculate bills for the service provider? If so, do you have their rates or do they interface their charges to you? Refer to Billing Relationships for more information.
- Do you send the customers' consumption to the service provider? Do they send it to you? Refer to Consumption Relationships for more information.
- How are financial settlements between your organization and the service provider implemented (do you pay them when you get paid?, do you purchase the receivable for them?, etc.). Refer to Service Providers Have Service Agreements Too for more information.
- Etc.

●

FASTPATH:
Refer to Designing Your SA Relationship Types and Service Providers for more information.

➤

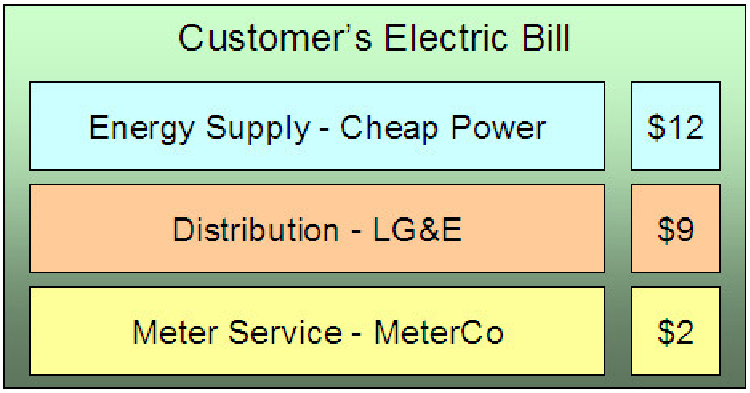
NOTE:
In some situations, you will need to set up a service provider for your own company. Refer to When Your Company Is A Service Provider for the details.

Service Providers Are Linked To Service Agreements

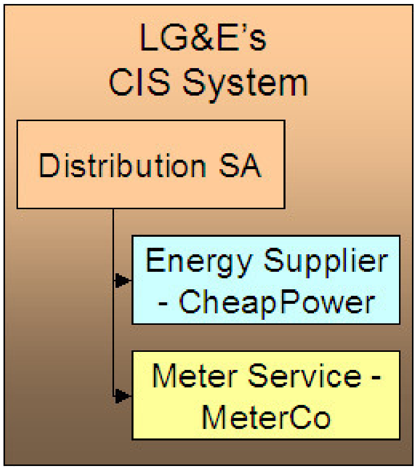
The following diagram illustrates a customer's bill for electric service in a deregulated market. Notice that there are separate sections for energy, distribution and meter service.

➤

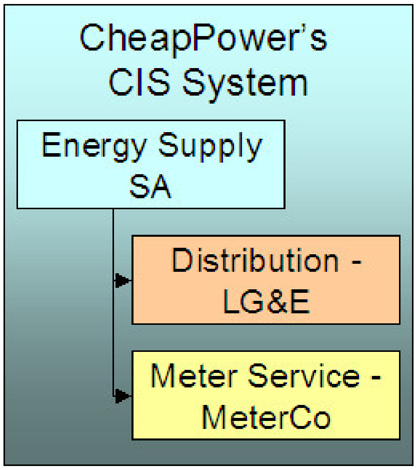
NOTE:
Consolidated billing. The following is an example of a bill that consolidates charges from many service providers. Rather than receive a consolidated bill, it is possible for the customer to receive 3 separate bills, one from each service provider (we refer to this as Dual billing).



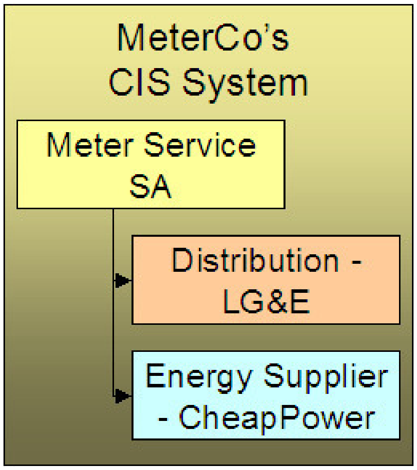
If we were to look at this customer's service agreement in the distribution company's CIS system, we'd find a service agreement for distribution charges, and linked to it would be information about the meter service and energy service providers:



If we look at this customer's service agreement in the energy supply company's CIS system, we'd find a service agreement for energy charges, and linked to it would be information about the distribution and meter service providers:



And finally, if we look at this customer's service agreement in the meter service company's CIS system, we'd find a service agreement for meter service charges, and linked to it would be information about the distribution and energy providers:



NOTE:

Bottom line. A customer's service providers keep track of the customer and each other in their respective CIS systems. A customer will have a service agreement (or the equivalent) in each service provider's CIS system. A customer's service agreement defines the service providers who supply each type of service.

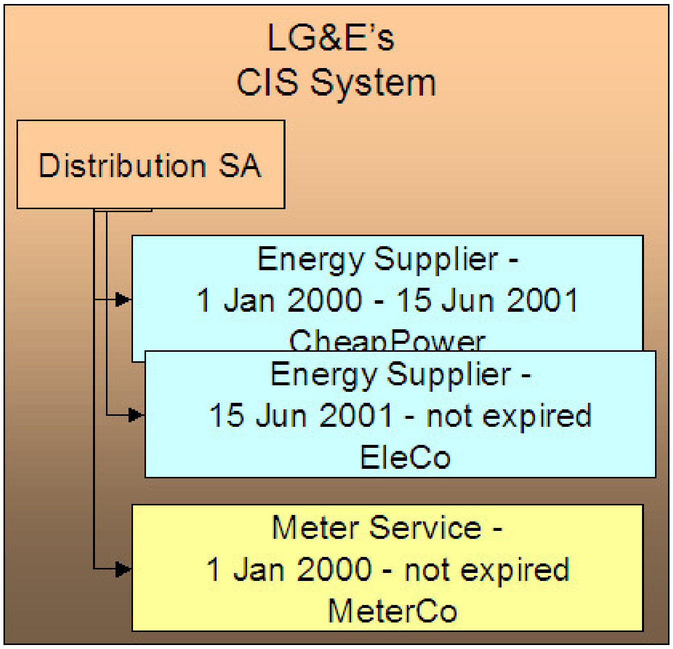
FASTPATH:

Because information about customers and their services needs to be kept up-to-date in many different CIS systems, there is a need for automated communications between service providers. Refer to Service Providers Have To Communicate About Customers for more information.

Service Providers May Change Over Time

A customer typically has a choice of service providers. Over time, their choice may change. The system keeps track of a customer's service providers throughout time so that it can accurately cancel / rebill historical bills. This means that the service provider relationship is *effective-dated*.

The following diagram illustrates how a distribution company's system keeps track of a customer's service providers (notice the customer changed energy suppliers):



How To Set Up SA Relationships On A Customer's Service Agreement

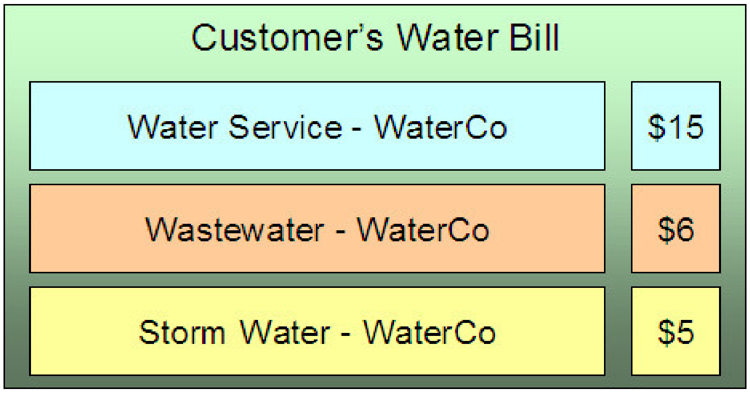
There are three ways to set up a customer's service providers:

- **Manually.** An operator can manually change a service agreement's SA relationships. The manual method is NOT recommended as changing service providers typically involves many events (e.g., you have to notify the current service provider that they will be dropped).
- **Use Default Service Providers.** The system will default a service provider on a required SA relationship when a service agreement is activated. Refer to Defaulting Relationship Types And Defaulting Service Providers for more information.

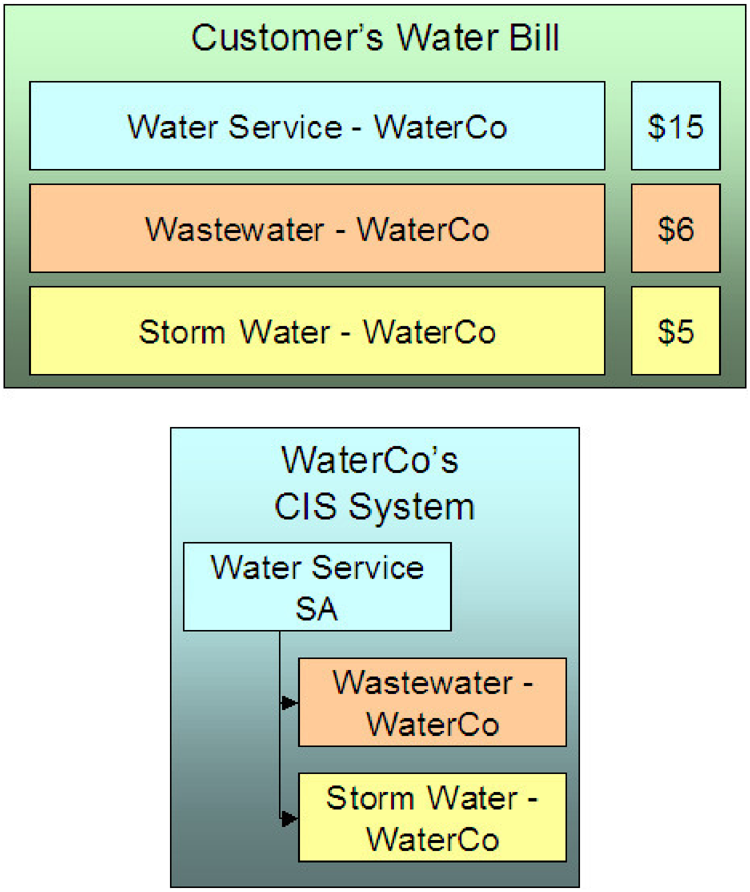
When Your Company Is A Service Provider

Besides setting up service providers for other companies that provide services to your customer, you may have to set up a service provider for your own company. You have to do this when:

- **Your organization can supply one of the services.** Refer to the illustration in Service Providers Are Linked To Service Agreements. If you are LG&E and you supply energy in addition to distribution, you would need to set up a service provider for your own organization. Why? Because whenever you have a subcategory of service (e.g., energy supply), you must indicate the service provider who provides this service; even when it's you.
- **You decide to break up a service into subcategories** (and have a separate service agreement for each category). For example, a water company may choose to break up service charges into water, wastewater and storm water (they may do this because there are different rates for each category of service). The following is an example of the segregated charges associated with this water company's service charges.



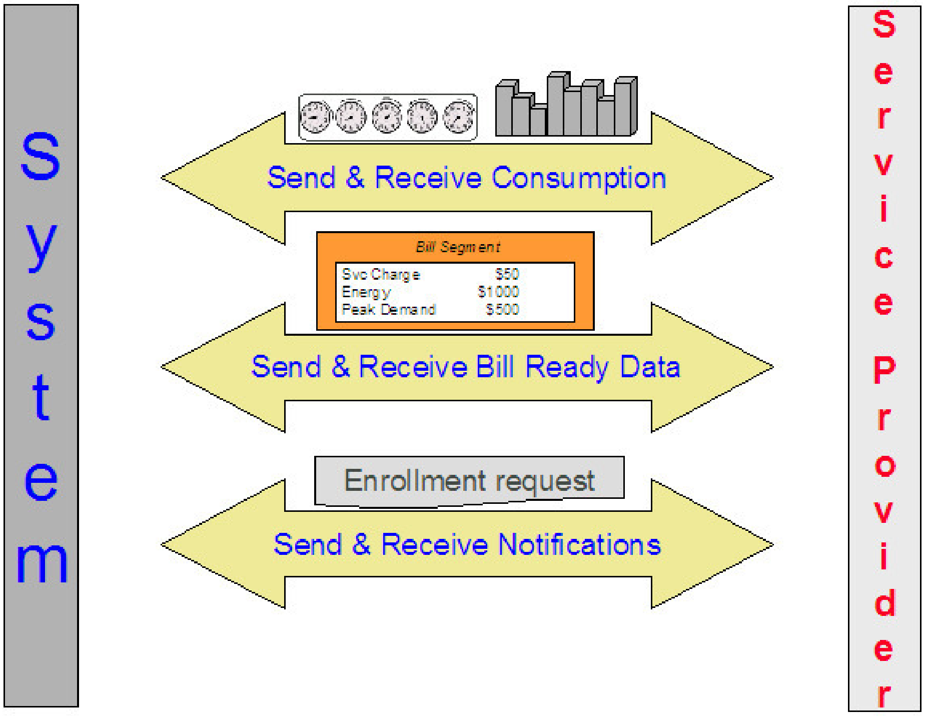
This water company system may benefit by creating a single service (for water) and indicating there are subcategories of service (for wastewater and storm water). Whenever you have a subcategory of service, you must indicate the service provider who provides this service. And, in this example, the water company would be the sole service provider for each subcategory of service.



NOTE:
Refer to We Bill For Them - Rate Ready Bill Segments Are Special for restrictions in respect of using subcategories of service.

Service Providers Have To Communicate About Customers

The providers of service typically have to communicate with each other in respect of the customer's service. The following diagram illustrates the major interfaces of information between your system and your service providers.



Depending on where your organization fits in the service provider hierarchy, you may:

- Bill for other service providers (or they may bill for you). Refer to Billing Relationships for more information about billing communications.
- Send the customer's consumption to service providers (or they may send it to you). Refer to Consumption Relationships for more information about consumption communications.
- Apprise service providers of the changes to the customers' service (or they may apprise you). Refer to How Do You Communicate With Service Providers? for more information about communications between service providers.

Relationships Between Service Providers

Service providers may arrange contractual relationships with other service providers to provide additional services, e.g., an energy service provider may work with a specific meter data management agency to gather and report interval meter read data.

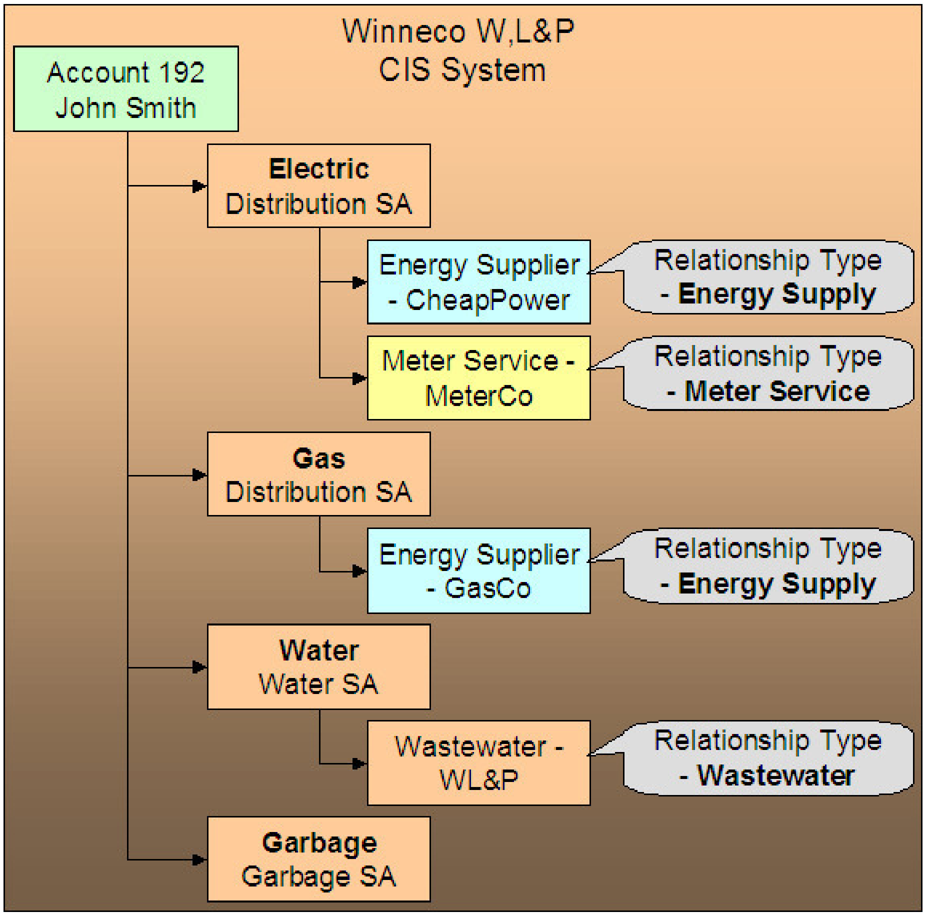
The system does nothing special to enforce or record these inter-relationships. Why? Because most service provider switches are received by notification records. Notification records indicate all the associated service providers. There is no need for the system to maintain the inter-service provider associations.

A Service Agreement Can Have Many Types Of Relationships

As described in the previous sections, a given service can be subdivided into subcategories. Each type of service can have zero, one or more subcategories. We call each subcategory a **SA Relationship Type**. The topics in this section provide information about SA relationship types.

An Example

We'll use the following example of a customer in a municipal utility's CIS system to explain SA Relationship types:



Note the following:

- Electric service has 2 SA relationship types: energy supply and meter service.
- Gas service has 1 SA relationship type: energy supply.
- Water service has 1 SA relationship type: waste water.
- Garbage service has no SA relationship types.

NOTE:

Bottom line. Service providers are related to the customer via "your" service agreement. Each service provider linked to a service agreement is defined in respect of a SA Relationship Type. This relationship is effective-dated because we care about how it changes over time.

Your relationship type is implied. The SA relationship type of "your" service agreement is implied, e.g., if you are a distribution company, "your" service agreement's implied SA relationship type is "distribution".

Valid Relationship Types and Service Providers Are Defined On SA Types

You control which services have SA relationships (and which don't) when you set up your SA types. Each SA type can have zero, one or more SA relationship types. Each relationship type, in turn, can have one or more valid service providers.

FASTPATH:
Refer to Setting Up SA Relationships For SA Types for the page used to define the service providers and SA relationship types for each SA type.

Defaulting Relationship Types And Defaulting Service Providers

Please be aware of the following:

- A SA relationship type for a given SA type can be marked as being **Required**.
- A service provider for a given SA relationship type / SA type can be marked as being the **Default**.

If, at activation time, the customer's master SA is missing a **Required** SA relationship that has a **Default** service provider, the activation process automatically creates the SA relationship type and links to it the **Default** service provider. If a master service agreement doesn't have all **Required** SA relationships, the service agreement cannot be activated. This is handy when your organization is the default service provider for a relationship type.

FASTPATH:
Refer to Automatic Creation of Sub SAs for information about how the system will automatically create sub service agreements for the defaulted SA relationship / service provider if your organization provides billing services for the service provider. Refer to Setting Up SA Relationships For SA Types for the page used to define the service providers and SA relationship types for each SA type.

Required Relationship Types and Billing

When the system attempts to create a bill segment for a service agreement whose SA type has **Required** relationship types, it checks if all such relationships are defined for the service agreement. If not, a bill segment error will be generated.

The reason this restriction exists is to handle the situation when your required relationship types change over time. For example, assume on your first day of production you only need energy suppliers defined on electric service agreements. After several months, gas deregulates. When this happens, you will need to change your control tables to indicate that your gas SA types require an energy supplier. If you don't write a default program to update your existing gas service agreements, billing will complain.

Relationship Types Do Not Impact Start/Stop

Customer service representatives (CSR's) are typically not involved with the customer's choice of service providers. Most organizations hear about a customer's service providers from the service providers or from a central body. This means that the start/stop dialog is not impacted by SA relationships. This also means that CSR's are not impacted by SA relationships (unless something goes wrong). If something goes wrong, the CSR's may need to manually correct SA relationships. Refer to How To Set Up Service Providers On A Customer's Service Agreement for more information.

Billing Relationships

When you set up a service provider, you must define your organization's billing relationship with the service provider. The following points provide examples of the billing relationships supported in the system,

- If you are an energy supply company, you may provide billing services for the distribution company. This means that your bill contains both your charges and the distribution company's charges. We refer to this as the **We Bill For Them** billing relationship.
- Alternatively, the distribution company may provide billing services for you. This means that the distribution company's bill contains your charges and their charges. We refer to this as the **They Bill For Us** billing relationship.
- Alternatively, you may both send bills to the customer. We refer to this as the **Dual Billing** relationship.
- Alternatively, if you subcategorize your services OR if your company provides one of the services that is provided by your service providers, then the system will create a separate bill segment for the subcategory of service. We refer to this as the **It's Us** billing relationship.

If you provide billing services for another service provider (i.e., you bill for them), there are two ways to determine the service provider's charges:

- You can load the service provider's rates in your system and calculate the charges for the service provider. We call this the **Rate Ready** calculation method.
- You can let the service provider calculate their own charges and interface them to you. We call this the **Bill Ready** calculation method.

If a service provider provides billing services for you (i.e., they bill for you):

- If the service provider has a suitable CIS system, they can load your rates in their system and calculate your charges for you. We call this the **Rate Ready** calculation method.
- You can calculate your charges and interface them to them. We call this the **Bill Ready** calculation method.

If you don't have a billing relationship with a service provider, you still need a service provider record to define such. Why? Because the system needs to know that it doesn't have to worry about a particular service provider in respect of billing. In addition, you may have other interactions with a service provider that have nothing to do with billing, e.g., you may send or receive consumption.

The topics in this section provide a wealth of information about the various billing relationships and the ramifications of each.

Sub Service Agreements

If you provide billing services for another service provider OR if you subcategorize your own charges, there will be a separate service agreement (SA) linked to the customer’s account that holds these unique charges. We refer to this new service agreement as a **Sub SA**. We use the term “sub” because this service agreement is subservient to the “master” service agreement. By subservient we mean:

- A sub SA’s start and stop dates are the same as the master SA. This statement may seem odd, but it’s true – all sub SAs linked to a master service agreement have the same start and stop dates as the master.
- A sub SA’s status (i.e., *pending start*, *active*, *pending stop*, *stopped*, etc.) is controlled by its master service agreement. As a rule, the master SA transitions its status first and the sub SA(s) follow.

➤ **NOTE:** There are situations where the master service agreement transitions its status and the sub service agreement(s) does not. One example of this is when both the master and sub SAs are closed and a payment cancellation occurs that only affects the master SA. In this scenario, the master SA becomes reactivated and the sub SA remains closed. Another example is when a straggling billable charge posts to a sub SA; the master SA is not affected.

- Refer to The Rate Ready Calculation Method for additional examples of subservience.

➤ **NOTE:**
You may find it helpful to keep in mind that sub SAs are only used for service providers with a billing relationship of It's Us and We Bill For Them. This is because these are the only relationships that have implicit billing responsibilities.

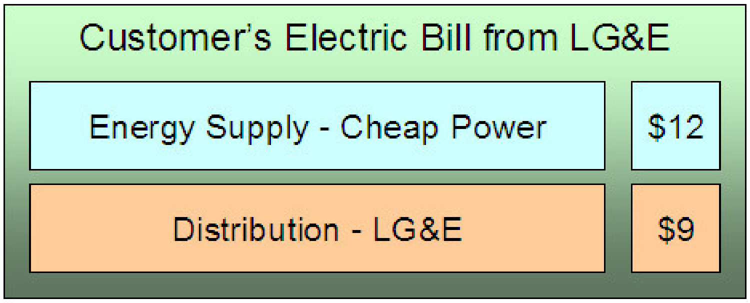
The topics in this section provide additional information about sub SAs.

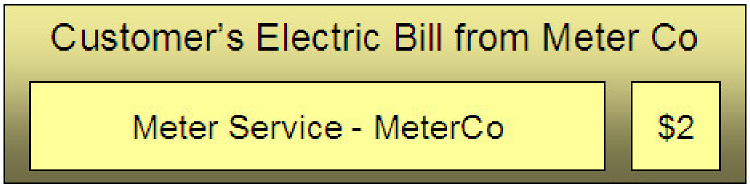
Only Some Service Providers Have Sub SAs

If you do not provide billing services for a service provider, there will be no sub SAs associated with the service provider's SA relationships. Let's use an example to make the point; assume:

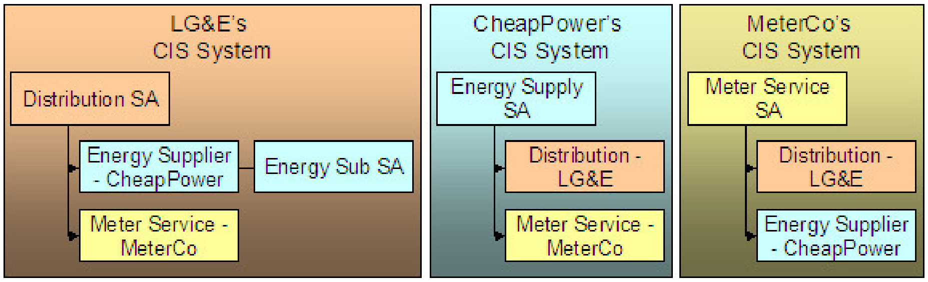
- Service is divided into distribution, energy and meter service charges.
- You are the distribution company and you provide billing services for the energy supply company.
- The meter service company bills the customer independently.

In this situation, the customer will receive two bills: one from you (LG&E), the other from MeterCo. Notice that your bill (LG&E) contains your distribution charges AND CheapPower's energy charges:


Customer's Electric Bill from LG&E
Energy Supply - Cheap Power
$12
Distribution - LG&E
$9



If we were to look at the customer information in the three service providers' respective CIS systems, we'd find the following service agreements and SA relationships for the customer:



Notice the following:

- The customer has a service agreement in each supplier's CIS system.
- Because LG&E (the distribution company) bills for CheapPower, all customers who have their energy supplied by CheapPower will have a sub SA in LG&E's system. This sub SA maintains the charges (and receivable balance) associated with CheapPower's energy charges.
- Notice that neither CheapPower's nor MeterCo's CIS systems use sub SAs. This is because neither company bills for other service providers.

NOTE:
Sub SAs are needed if you subcategorize your charges. The above example shows sub SAs being used when a company provides billing service for another company. Sub SAs are also used when you subcategorize your charges - each sub SA contains the rate associated with each subcategory.

Automatic Creation of Sub SAs

The system creates sub SAs for customers choosing service providers where the billing option is It's Us or We Bill For Them. The system creates sub SAs via the following mechanisms:

- The analyze SA relationships background process (ANLYZSAR) monitors newly activated SA relationships. If the respective service provider is It's Us or We Bill For Them, this process creates the sub SA(s) using the information defined on SA Type SA Relationship Type - Sub SA Type.

NOTE:
The analyze SA relationships background process also activates and stops sub SAs. Refer to Sub SA State Transition for the details.

- A button exists on the SA Relationships page. This button, when pressed, creates sub SAs real-time. This button would only be used if the operator couldn't wait for the background process to run.

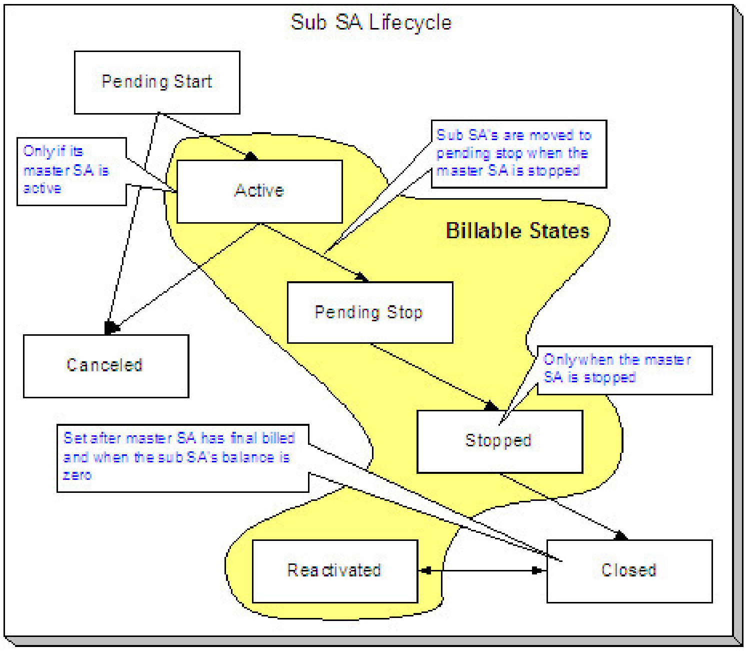
The SA type associated with the new sub SAs is defined when you set up each SA Type / SA Relationship Type. Refer to Setting Up SA Relationships For SA Types for more information.

If the SA type uses start options, the start option contains additional values (e.g., rates, contract riders) that are used to populate the newly created sub SAs. Note: start options are typically not used for We Bill For Them - Bill Ready service provider because we don't need to default rates, contract riders and/or contract values on billable charge sub SAs. Refer to Setting Up SA Relationships For SA Types for more information.

NOTE:
Manually created sub SAs. In very unusual situations, an operator may create a sub SA manually. An operator would do this using the SA Relationship transaction.

Sub SA State Transition

Sub SAs follow normal SA state transition rules, but have a few additional rules about when the transitions can take place.


Sub SA Lifecycle
Pending Start
Only if its master SA is active
Active
Sub SA's are moved to pending stop when the master SA is stopped
Billable States
Pending Stop
Canceled
Only when the master SA is stopped
Stopped
Set after master SA has final billed and when the sub SA's balance is zero
Reactivated
Closed

The following points highlight the additional state transition rules:

- A sub SA can only become Active if its master SA is Active. Most sub SAs are activated by the analyze SA relationships background process. However, a user can manually activate a sub SA using the activate button on the SA maintenance page.

- Sub SAs become Pending Stop when their master becomes Stopped . This typically occurs when the SA activation background process stops the master SA. However, this can happen real time if a user manually stops a master SA using the stop button on the SA maintenance page.
- A sub SA can only become Stopped if its master SA is Stopped . Most sub SAs are stopped by the analyze SA relationships background process. However, a user can manually activate a sub SA using the stop button on the SA maintenance page.
- A sub SA becomes Closed if its master has been final billed and it has a balance of zero. Most sub SAs are closed when their balance becomes zero after the master SA is final billed. However, a user can manually close a sub SA using the close button on the SA maintenance page. Note, if additional billable charges are interfaced after the master has been closed, the sub SA will be Reactivated.

We Bill For Them

If you provide billing services for another service provider, then you have a We Bill For Them billing relationship with the service provider. The topics in this section provide information about this type of billing relationship.

Sub SAs Are Used When We Bill For Them

As described under Sub Service Agreements, sub SAs are used for We Bill For Them service providers. The sub SAs hold the service providers charges. It's important to note that the system allows more than one sub SA to be created for a given service provider.

The Rate Ready Calculation Method

If you provide billing services for a service provider (i.e., the service provider has a billing relationship of We Bill For Them), you can load the service provider's rates in your system and calculate the charges for the service provider. We call this the **Rate Ready** calculation method.

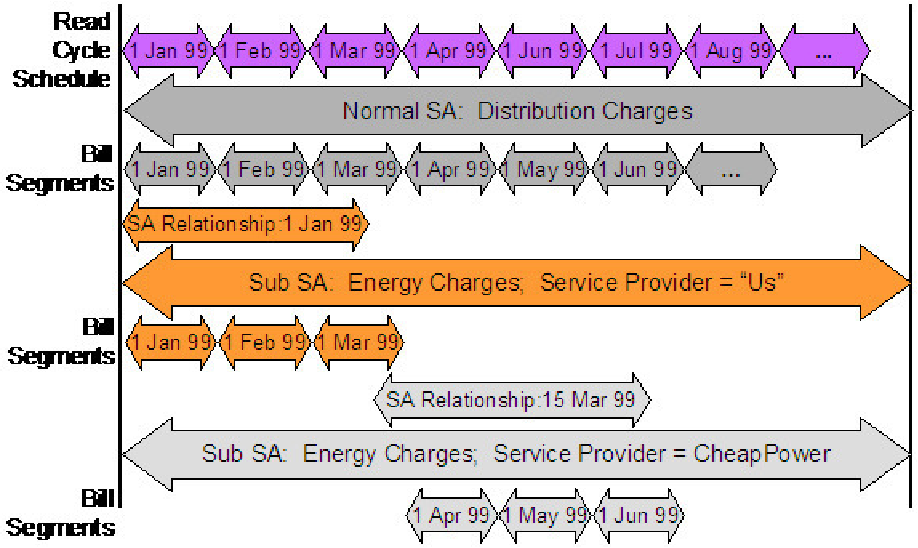
> **NOTE:**
> It's Us **can be Rate Ready.** The prior paragraph indicated that the Rate Ready billing option was used for We Bill For Them service providers. If you subcategorize your services and use rates to calculate each subcategory's charges, the service provider(s) set up for your own organization will also be subject to the Rate Ready rules described below.

If a customer uses a We Bill For Them - Rate Ready service provider, the system will create a separate bill segment for the service provider's charges. Keep in mind the following in respect of these types of bill segments:

- The consumption is copied from the "master" service agreement (note: we copy item details and read details after pre-processing calculation groups, if any exist, have been applied). This means that the bill period on the sub SA's bill segment is identical to that on the normal SA. This also means that if the "master" SA is in error, bill segments for the sub SA will not be created.
- The rate defined on the sub SA is applied against this consumption. All pre-processing calculation groups defined on the service provider's rate will be used to manipulate this consumption. If you have pre-processing calculation group on your master SA's rate to apply a line loss or convert a cubic foot to a therm, you don't have to have these pre-processing

calculation group on the sub SA's rate (because we copy the consumption from the master to the sub after pre-processing calculation group, if any exist, have been applied).

The following illustration should help:



Be aware of the following:

- This example is from a distribution company's perspective.
- From 1 Jan 99 through 15 Mar 99, the distribution company distributes power AND supplies energy (i.e., the distribution company is the service provider of energy).
- On 15 Mar 99, CheapPower becomes the energy supplier. However, the next bill period ends on 30 Mar 99. This means that CheapPower will only start supplying energy on 1 Apr 99. See the following note for why the effective date of the supplier switch is not the true effective date from billing's perspective.

> **NOTE:**
> **Service Provider Changes Take Effect On The Next Bill.** If the service provider is changed within a customer's billing period, the system assumes that the service provider in effect at the START of the period is effective the entire billing period. This means that a change of service providers will only take effect on the bill whose start date follows the change date.

Please be aware of the following characteristics of Rate Ready bill segments:

- You cannot cancel a Rate Ready bill segment independently from the master bill segment. If you need to cancel / rebill the sub SA's bill segment, you must cancel / rebill the master SA's bill segment.
- You cannot delete a Rate Ready bill segment independently from the master bill segment. If you need to delete the sub SA's bill segment, you must delete the master SA's bill segment.
- Similarly, if you cancel / rebill the normal SA's bill segment, all Rate Ready bill segments will be cancelled / rebilled.

> **NOTE:**

> **Bottom line.** Creating, freezing, deleting, and canceling a "master" bill segment does the same with all Rate Ready bill segments. You cannot create, freeze, delete or cancel a Rate Ready bill segment independent of its master bill segment.

The Bill Ready Calculation Method

If a customer uses a We Bill For Them - Bill Ready service provider, the service provider calculates their own charges and interfaces them to the system. We then present their charges on our bill; we don't actually calculate anything (sometimes this is referred to as "pass through" billing).

The sub SA linked to a We Bill For Them - Bill Ready must be a billable charge SA. Why? Because billable charge SAs exist to hold bill lines until such time as a bill is created for the customer's account. Refer to Setting Up Billable Charge Templates for more information.

Please be aware of the following characteristics of Bill Ready bill segments:

- Unlike Rate Ready bill segments, Bill Ready bill segments can span different time periods than the master SA. This is because you cannot predict when a service provider will interface their billable charges to you. In fact, a given bill could contain billable charges that span different periods and these charges could have been interfaced from historic and existing service providers.
- Unlike Rate Ready bill segments, Bill Ready bill segments can be created and deleted independently from the master bill segment.
- Unlike Rate Ready bill segments, Bill Ready bill segments can be cancelled independently from the master bill segment.

Sending Consumption And Waiting For The Charges

Uploading Consumption (Rather Than Uploading Calculated Charges)

Calculating Taxes On Uploaded Charges

Sending Consumption And Waiting For The Charges

If your organization supports We Bill For Them - Bill Ready service providers AND you are the source of consumption used by these service provider to calculate their charges, you need to be aware of the following:

- We do NOT recommend sending raw meter reads to service providers. Rather, we recommend sending these service providers the same consumption that you use on your bill segments. Remember, the system maintains a snapshot of billed consumption on bill segments associated with service agreements that are linked to service points.
- But to implement our recommendation (of only interfacing billed consumption to service providers), we need to create a bill segment for the master SA and then wait until the service provider returns the billable charge before sending out the bill. The following points describe how this works:
  - Early in the bill cycle, the system creates a bill segment for the master SA (remember, the system maintains a snapshot of billed consumption on bill segments linked to service points). When the master bill segment is frozen, the system interfaces the snapshot consumption to all service providers associated with the master service agreement who need consumption (this is defined on the service provider). Refer to We Can Send Billed Consumption To Any Service Provider for more information about interfacing consumption to service providers.
  - The bill associated with the bill segment will not be completed. Why? Because the bill segments associated with the We Bill For Them - Bill Ready service providers will be in Error. This only happens if you use the Billable Charge bill segment creation algorithm (this algorithm is plugged in on the sub SA's SA type). On the algorithm, make sure to specify a value of Y for the parameter **Wait For The Last Day Of The Bill Cycle**. By doing this, a bill segment in the Error state will exist for the sub SA until the last night of the bill cycle's window.
  - On the last night of the bill cycle window, when the system attempts to create a bill segment for the billable charge SA associated with the We Bill For Them - Bill Ready, it will either find recently interfaced billable charges or it

won't. If it finds unbilled, billable charges, a bill segment will be created for the sub SA and the billable charges will be swept onto it. If it doesn't find unbilled billable charges, the bill will be completed (i.e., sent out) without the service provider's charges.

> **NOTE:**
> **Bottom line.** If a customer's account uses We Bill For Them - Bill Ready service providers, the bill will not be completed (i.e., sent out) until the last night of the bill cycle (if you use the appropriate algorithm). Why? Because we wait until the last night of the bill cycle before trying to sweep on recently interfaced billable charges. If no billable charges have been interfaced from the service provider by the last night of the bill cycle, the bill will be sent out without the service provider's charges.
>
> **Batch versus Online Bill Creation.** If you create a bill online, the system will NOT create an Error bill segment for the We Bill For Them - Bill Ready service provider. Why? Because if you want to create an online bill, either the service provider has interfaced their charges or they haven't. If they have, they should be swept on the bill (via the creation of a bill segment). If they haven't, it shouldn't prevent you from completing the bill.

Top of the Page

Uploading Consumption (Rather Than Uploading Calculated Charges)

We understand this is confusing, but it is quite possible to set up the system so that the We Bill For Them - Bill Ready service provider passes in CONSUMPTION rather than the calculated bill lines. They would only do this if they are not able to calculate the charges in their system and have therefore provided you with their rates. To do this, you would set up everything as described above. In addition, when you upload the billable charges, you must specify the consumption to be rated in the billable charge's service quantity (SQ) collection.

Top of the Page

Calculating Taxes On Uploaded Charges

It is possible to set up the system to calculate taxes for billable charges. You would do this if the service providers are passing through the charges and want you to calculate the taxes.

If you want taxes calculated on top of billable charges:

- Create a service quantity (SQ) on the billable charge that contains the total monetary amount that taxes will be calculated on. Note, you would not have to do this if you have a pre-processing calculation group in your rate that calculates the total monetary amount to which taxes should be applied.
- Specify a rate on the billable charge service agreement. This rate will contain calculation rules that calculate taxes. Note, the calculation rules will be simple SQ calculation rules that apply a percentage to the value of the SQ that represents the monetary amount on which taxes should be calculated.

When the system creates the bill segment for the billable charge, it will call the rate and the rate will calculate the taxes and add additional lines (actually, bill calculation headers) to the bill segment.

Pay At Bill Time vs. Pay At Pay Time

If you provide billing services for another service provider (i.e., the service provider has a billing relationship of We Bill For Them), you will owe them money because you will be receiving money from their customers for their service. You have two options in respect of when the system increases the amount you owe the service provider:

- You can tell the system to increase how much you owe the service provider when you freeze the customer's bill segment. Some people refer to this method as "purchasing the receivable from the service provider". We call this the **Pay At Bill Time** method.

- You can tell the system to increase how much you owe the service provider only when you are paid for by the customer. We call this the **Pay At Pay Time** method.

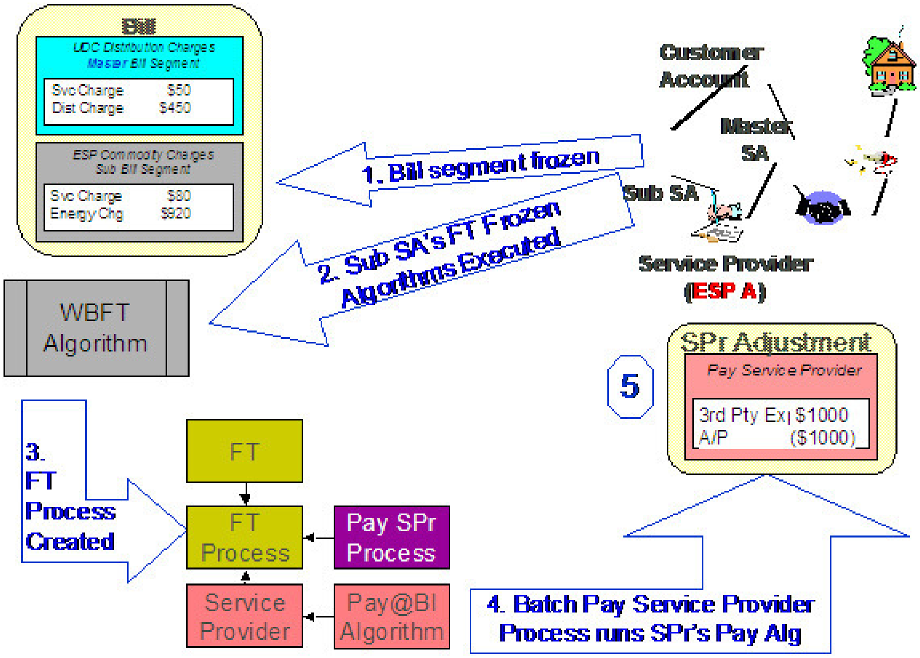
The method used for a service provider is defined on the service provider's record.

- **FASTPATH:**
  Refer to When We Bill For Them, We Owe Them Money for more information.

Paying The Service Provider - Technical Implementation

WARNING:
This section describes, technically, how we increase the amount we owe a We Bill For Them service provider. If you aren't technically inclined, skip this section.

The following illustration shows the logical steps involved with increasing how much we owe a We Bill For Them service provider.



The following points explain the steps:

- When a financial transaction (FT) is frozen, the system executes the FT Freeze algorithms defined on the SA(s) SA type.
- If you've set up the system properly (i.e., you've put the appropriate FT Freeze algorithm on the sub SA's SA type), one of these algorithms will determine if there is a WBFT service provider associated with the sub SA. If so, it will insert a row on the FT Process table.
- Rows on the FT process table are used as "triggers" for batch processes. In this case, the batch process that is triggered is the one that looks at new FT's and determines if a related "payable" adjustment should be created for the We Bill For Them service provider. This batch process uses the service provider's Payment Relationship and Pay Service Provider algorithm to determine when and how to create these "payable" adjustments.

Service Providers Have Service Agreements Too

Most service providers need service agreements as explained in the following topics.

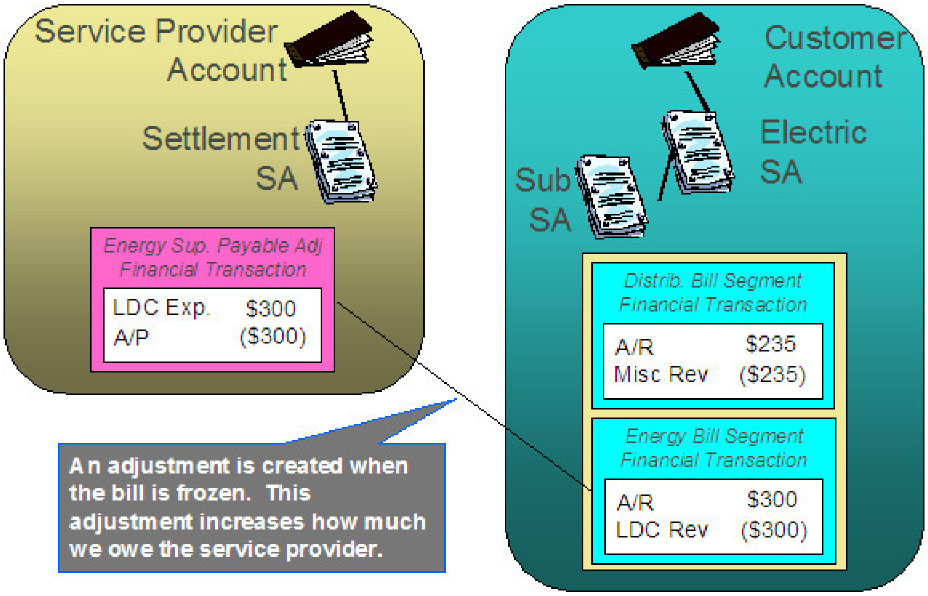
When We Bill For Them, We Owe Them Money

When you bill on behalf of a service provider (i.e., the service provider has a billing relationship of We Bill For Them), you will eventually owe them money (because the customer pays you for the service provider's service). You have two options in respect of when the system increases the amount you owe the service provider:

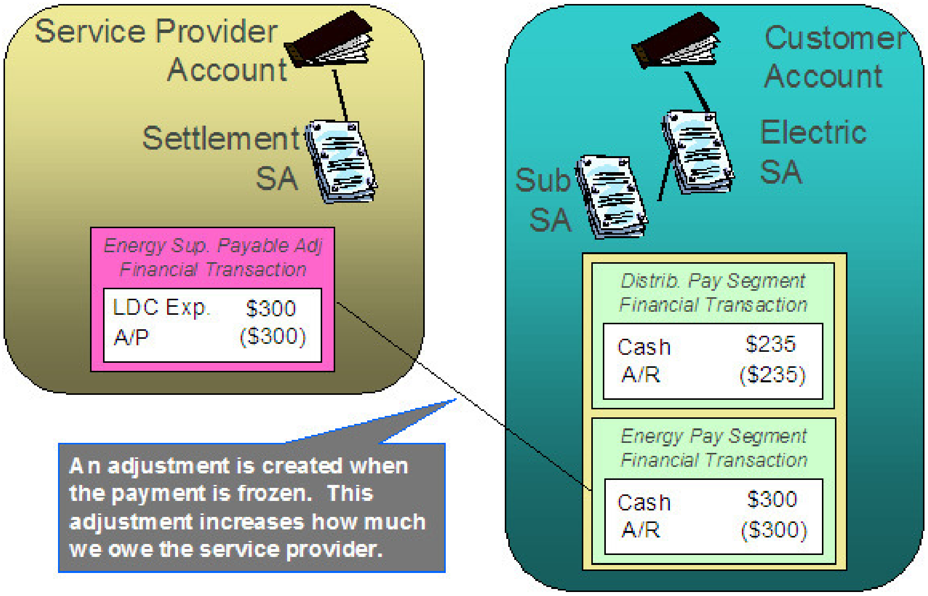
- You can tell the system to increase how much you owe the service provider when you create the customer's bill. Some people refer to this method as "purchasing the receivable from the service provider".
- You can tell the system to increase how much you owe the service provider only when you are paid for by the customer.

The system keeps track of how much you owe a service provider on a service agreement linked to the service provider's account. The system creates adjustments against this service agreement to increase how much you owe them.

If you "purchase the receivable" (i.e., you owe them when you bill the customer), an adjustment is created when the customer is billed. If you owe them only when you are paid by the customer, an adjustment is created when the customer pays. The following example illustrates an adjustment being created when the bill is frozen (illustrating the "purchase the receivable" scenario):


Service Provider Account
Settlement SA
Energy Sup. Payable Adj Financial Transaction
LDC Exp. $300
A/P ($300)
Customer Account
Electric SA
Sub SA
Distrib. Bill Segment Financial Transaction
A/R $235
Misc Rev ($235)
Energy Bill Segment Financial Transaction
A/R $300
LDC Rev ($300)
An adjustment is created when the bill is frozen. This adjustment increases how much we owe the service provider.

If you only pay the service provider when you are paid, the example would look as follows:


Service Provider Account
Settlement SA
Energy Sup. Payable Adj Financial Transaction
LDC Exp. $300
A/P ($300)
Customer Account
Electric SA
Sub SA
Distrib. Pay Segment Financial Transaction
Cash $235
A/R ($235)
Energy Pay Segment Financial Transaction
Cash $300
A/R ($300)
An adjustment is created when the payment is frozen. This adjustment increases how much we owe the service provider.

Adjustments and We Bill For Them Service Providers

- **FASTPATH:**
  It's important that you are comfortable with the information described under When We Bill For Them, We Owe Them Money before reading this section.

Adjustments associated with We Bill For Them sub service agreements are tricky. The following points describe how the system "pays" the related service provider when adjustments are issued against the customer's sub service agreement:

- For Pay At Bill Time service providers, most adjustments are treated just like bill segments, i.e., when the adjustment's FT is frozen, a payable adjustment is created for the respective service provider. The reason "most" is underlined in the previous sentence is because A/P adjustments (i.e., adjustments used to interface check requests to your A/P system) are excluded. Why? Because A/P adjustments are used to refund overpayments to the customer. Overpayments are purely between the customer and your company (you never transferred the overpayment to the service provider because it's associated with a Pay At Bill Time service provider).
- For Pay At Pay Time service providers, A/P adjustments are treated just like payment segments, i.e., when the adjustment's FT is frozen, a payable adjustment is created for the respective service provider. All other types of adjustments are ignored. Why? Because A/P adjustments are used to refund overpayments to the customer. Think of it like this - when the customer originally overpaid, you transferred this overpayment to the Pay At Pay Time service provider; therefore, when you refund the overpayment, you get to take the money back from the service provider.

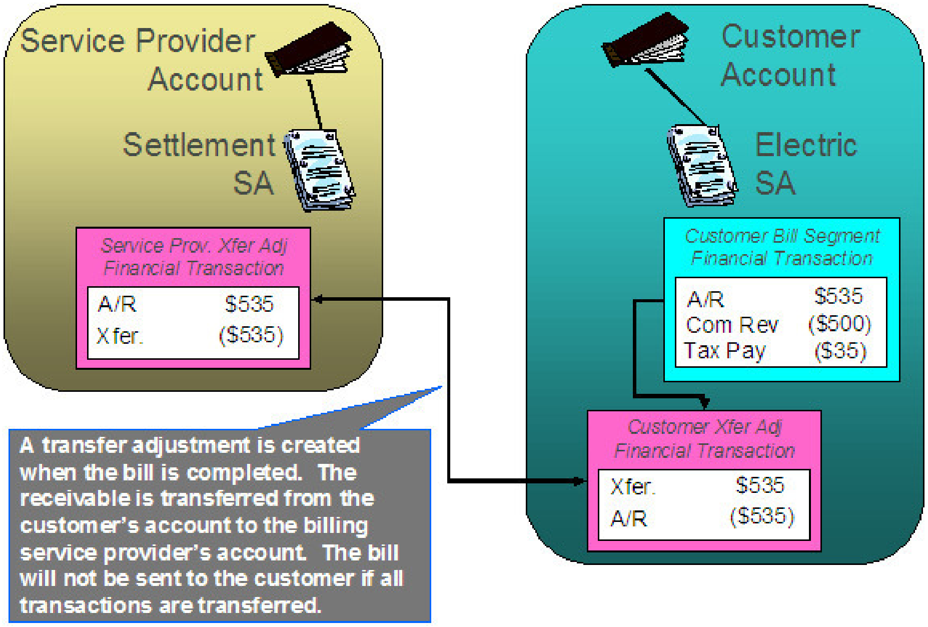
When They Bill For Us, They Owe Us Money

When a service provider bills on behalf of your organization, they will eventually owe you money (because the customer pays them for your service).

The system keeps track of how much a service provider owes you on a service agreement linked to the service provider's account. The question is, How does the system determine how much you are owed when you don't produce a bill? Well, you do produce a bill, it just doesn't get sent to the customer. We understand this is a little confusing, but think about it like this:

- The service provider is presenting your charges on their bill.
- You still have to calculate how much the customer owes your organization otherwise you'll never know how much you are owed by the service provider.

As illustrated below, when a bill is completed, the system determines if there are bill segments and/or adjustments associated with service agreements with a service provider who "bills for us". If it finds these, it transfers the receivable from the customer's service agreement to the service provider's service agreement. If all financial transactions have been transferred to the service provider, no bill is produced for the customer.


Service Provider Account
Settlement SA
Service Prov. Xfer Adj Financial Transaction
A/R $535
Xfer. ($535)
Customer Account
Electric SA
Customer Bill Segment Financial Transaction
A/R $535
Com Rev ($500)
Tax Pay ($35)
Customer Xfer Adj Financial Transaction
Xfer. $535
A/R ($535)
A transfer adjustment is created when the bill is completed. The receivable is transferred from the customer's account to the billing service provider's account. The bill will not be sent to the customer if all transactions are transferred.

NOTE:
Bottom line. We always generate a bill for "us", even though we don't send it to the customer.

They Bill For Us

If a service provider provides billing services for you, then you have a They Bill For Us billing relationship with the service provider. The topics in this section provide information about this type of billing relationship.

The Customer Still Needs A Service Agreement

If a service provider bills for you, you still need a service agreement for the customer. Why? Because:

- As explained under Service Providers Are Linked To Service Agreements, service providers are defined in respect of a customer's service agreement (therefore the customer must have a service agreement).
- As explained under When They Bill For Us, They Owe Us Money, you still have to calculate bills for the customer.

NOTE:
A customer's bill history still exists. Be aware that even when a service provider bills for us, you will still be able to see the customer's billing history. It's just that the customer won't owe you anything because the receivable balance will be transferred to the service provider's account.

They Bill For Us - Bill Ready

At bill completion time, the system determines if there are bill segments and/or adjustments associated with service agreements with a service provider who "bills for us". If it finds these,

- It transfers the receivable from the customer's service agreement to the service provider's service agreement. If all financial transactions have been transferred to the service provider, no bill is produced for the customer.
- Note that payments and A/P adjustments are not transferred. Why? Because payments and A/P adjustments are purely between the customer and your company.
- Each bill segment and adjustment is marked to be interfaced to the service provider (via a separate background process).

They Bill For Us - Rate Ready

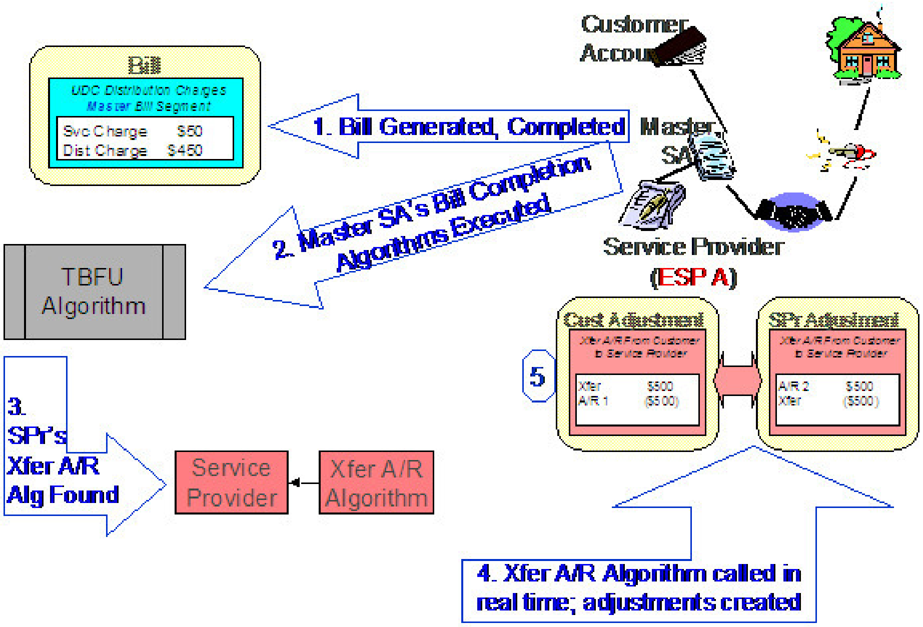
The They Bill For Us - Rate Ready option is not a recommended option. Why? Because you really have to compute how much the customer owes as explained above. If you go to the trouble of figuring out how much the customer owes, then it makes sense to interface this to the billing service provider.

The only difference between this option and They Bill For Us - Bill Ready is that the system will not interface the bill segments and adjustments to the billing service provider.

A/R Transfer - Technical Implementation

WARNING:
This section describes, technically, how a customer's A/R balance is transferred to a They Bill For Us service provider. If you aren't technically inclined, skip this section.

The following illustration shows the logical steps involved with the transference of a customer's A/R balance to a They Bill For Us service provider.


Customer Account
Bill
UDC Distribution Charges
Master Bill Segment
Svc Charge $60
Dist Charge $450
1. Bill Generated, Completed
Master SA
2. Master SA's Bill Completion Algorithms Executed
TBFU Algorithm
Service Provider
(ESP A)
Cust Adjustment
Xfer A/R From Customer to Service Provider
Xfer $500
A/R 1 ($500)
SPr Adjustment
Xfer A/R From Customer to Service Provider
A/R 2 $500
Xfer ($500)
5
3. SPr's Xfer A/R Alg Found
Service Provider
Xfer A/R Algorithm
4. Xfer A/R Algorithm called in real time; adjustments created

The following points explain the steps:

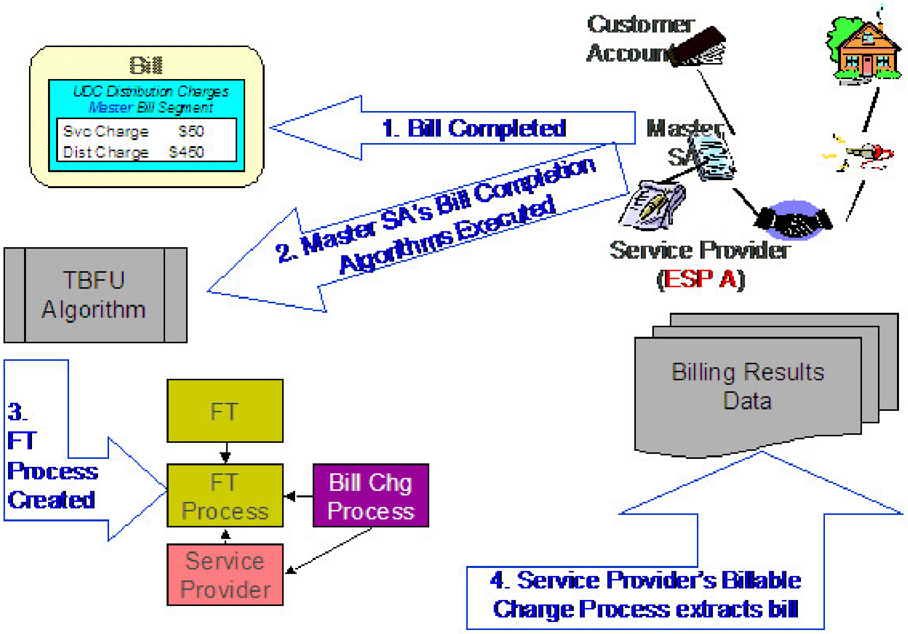
- When a bill is completed, the system executes the bill completion algorithms defined on the bill's master SA(s) SA types.
- If you've set up the system properly (i.e., you've put the appropriate Bill Completion algorithm on the master SA's SA type), one of these algorithms will determine if there is a They Bill For Us service provider associated with each master SA on the bill. If so, it will execute the Transfer A/R algorithm defined on the service provider's record. This algorithm causes a transfer adjustment to be created (transferring the financial transaction's affect on the customer's balance from the customer to the service provider).

NOTE:
If there are multiple master SAs on a bill, the financial details associated with each respective master SA could be transferred to different service providers (e.g., one service provider could receive the financial details for gas and another for electricity). Refer to Different Service Providers Can Bill Different Services for more information.

Routing Billable Charges To Service Providers - Technical Implementation

WARNING:
This section describes, technically, how we send billable charges to service providers. If you aren't technically inclined, skip this section.

The following illustration shows the logical steps involved with sending billable charges to service providers.



The following points explain the steps:

- When a bill is completed, the system executes the bill completion algorithms defined on the bill's master SA(s) SA types.
- If you've set up the system properly (i.e., you've put the appropriate Bill Completion algorithm on the master SA's SA type), one of these algorithms will determine if there is a They Bill For Us service provider associated with each master SA on the bill. If so, it will insert a row on the FT Process table.
- Rows on the FT process table are used as "triggers" for batch processes. In this case, the batch process that is triggered is the one that downloads billable charges to the service provider. The ID of the batch process that is referenced on the trigger comes from the Service Provider's Billable Charge Download Process.

NOTE:
If there are multiple master SAs on a bill, the financial details associated with each respective master SA could be routed to different service providers (e.g., one service provider could receive the financial details for gas and another for electricity). Refer to Different Service Providers Can Bill Different Services for more information.

Bill Routings Are Changed

If all of an account's "master" SAs have a SA relationship with a service provider who bills for us (i.e., the service provider's billing relationship is They Bill For Us), then we have nothing to send to the customer. The system still creates bill routings, but with a couple of differences:

- The Batch Process Id and Run Number are reset.
- The Customer's Name is prefixed with the text from a bill message code 6, 10103. This message code's text is *** Bill not sent.

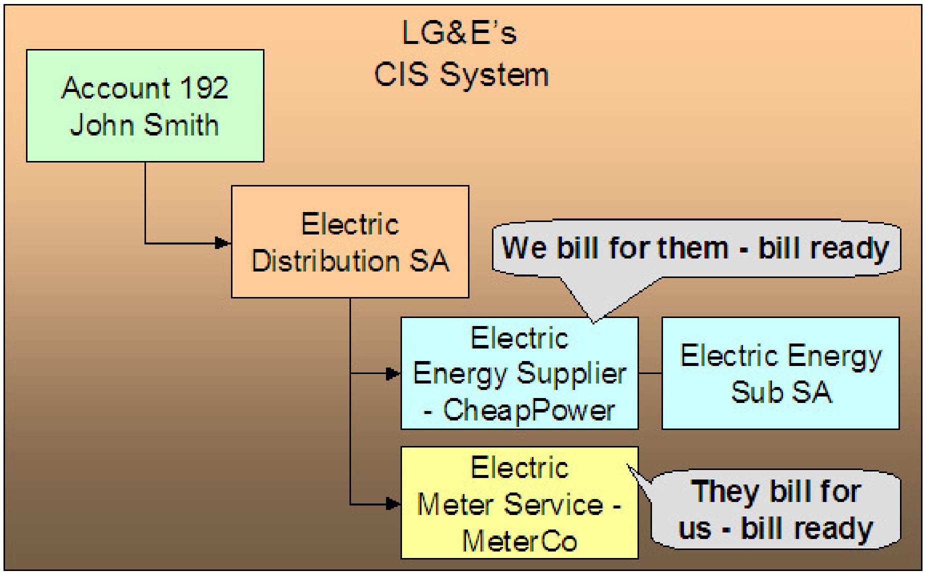
This way, the operators can easily see that that the bill was not routed and why.

FASTPATH:

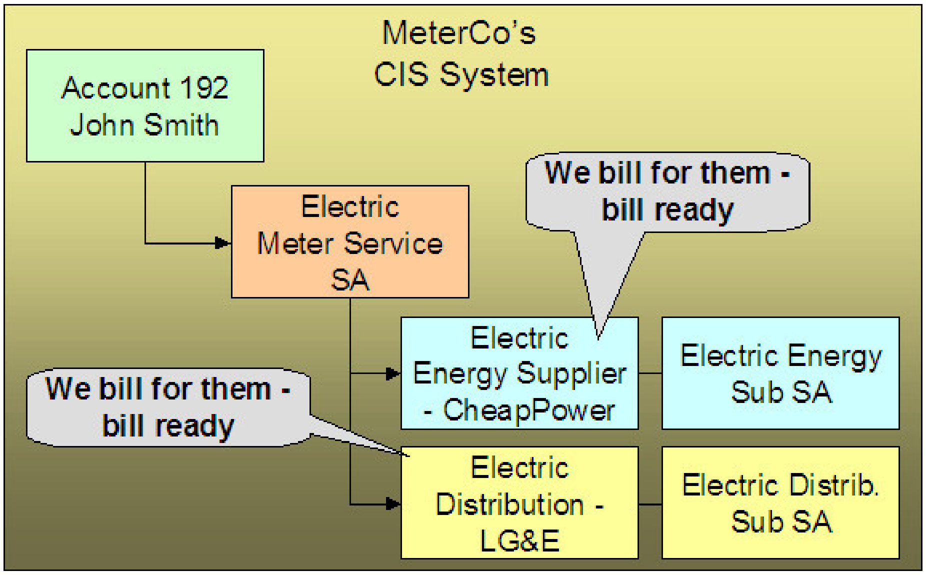
> Refer to Different Service Providers Can Bill Different Services for information about how the various services under an account could be billed by different service providers.

Combinations Of Service Provider Billing Methods

Consider the following situation:



In the above example, we are billing for the energy service provider and the meter service provider is billing for us. This means that we will calculate the charges for ourselves, interface charges from CheapPower, and then interface our charges and CheapPower's charges to MeterCo. MeterCo will then produce a bill for the customer that contains our distribution charges, CheapPower's energy charges, and MeterCo's service charges. To help solidify this point, let's look at how this customer would look in MeterCo's CIS system.

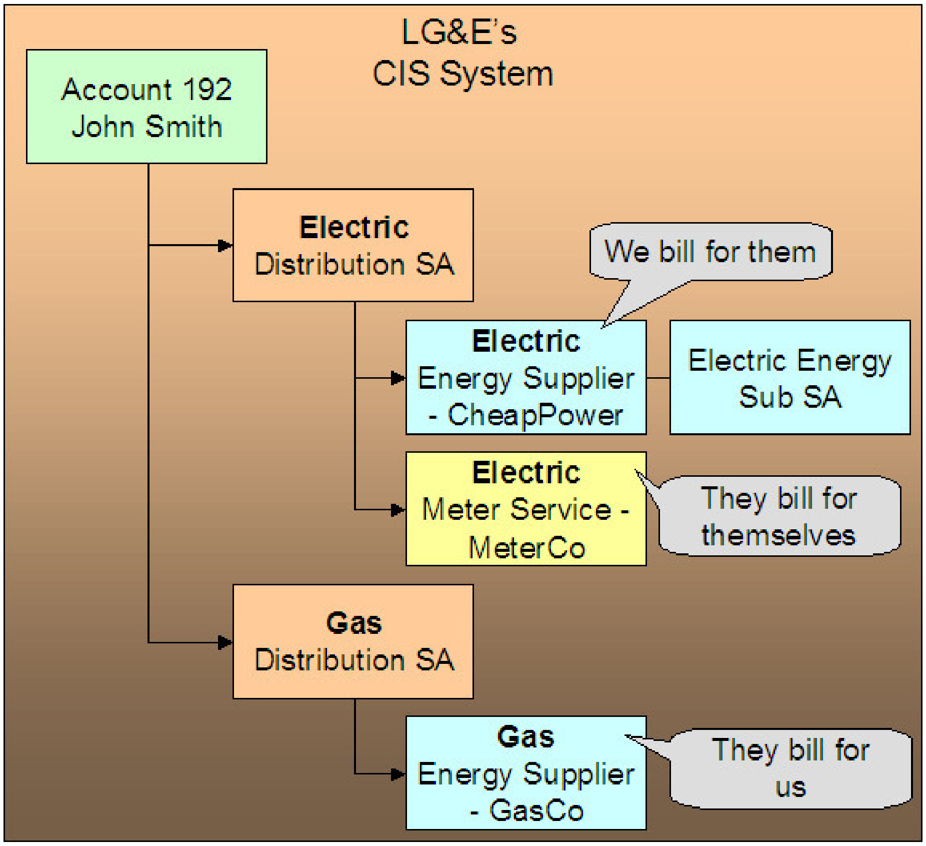


There are some restrictions in respect of permissible combinations of service providers that may supply service to a customer as described in the following points:

- If the system encounters a customer with a We Bill For Them (WBFT) - Bill Ready service provider and another service provider that is They Bill For Us (TBFU) - Rate Ready, a billing error will be produced. Why? Because TBFU - Rate Ready means they have everything they need to calculate our bills and therefore we do not interface bill lines to them. If we don't interface our bill lines to them, then we can't interface the charges that were interfaced from the WBFT - Bill Ready service provider. You may wonder why we don't prohibit WBFT - Rate Ready and TBFU - Rate Ready, because it's conceivable for the TBFU service provider to have our rate and the WBFT service provider's rate.
- If the system encounters a WBFT - Pay At Pay Time service provider and another service provider that is TBFU, a billing error will be produced. Why? Because when the system detects a TBFU service provider, it transfers the receivable from the customer to the service provider (and therefore the customer's account will never be paid).

Different Service Providers Can Bill Different Services

Be aware that the system determines billing relationships at the service agreement level, NOT at the account level. To make the point, check out the following customer in a distribution company's system:



Be aware of the following in respect of the above illustration:

- The distribution company (LG&E) distributes both electricity and gas.
- The customer has a choice of energy service providers for both gas and electricity.
- This customer - John Smith - purchases his electricity from CheapPower and his gas from GasCo.
- LG&E provides billing services for CheapPower.
- GasCo provides billing service for LG&E.

In this situation, LG&E will send bills to the customer that contain both electric distribution and energy charges (but no gas distribution charges). GasCo will also send bills to the customer; these will contain both LG&E's gas distribution charges as well as their own energy charges.

If You Deal With TBFU Service Providers, You Cannot Reopen Bills

● **FASTPATH:**
Refer to Bill Lifecycle for information about reopening previously completed bills.

If your organization deals with They Bill For Us (TBFU) service providers, a great deal happens when a bill is completed (e.g., the receivable is transferred from the customer to the service provider, we may mark the bill segments and adjustments for routing to the service provider, etc.). These things cannot be undone and therefore the system will not let you reopen bills when these things have occurred.

➤ **NOTE:**
Technical rule. The specific rule that prevents the reopening of bills is as follows: if a bill contains a service agreement whose SA type has one or more bill completion algorithms, the system will not allow the bill to be reopened. Refer to SA Type - Algorithms for more information about bill completion algorithms.

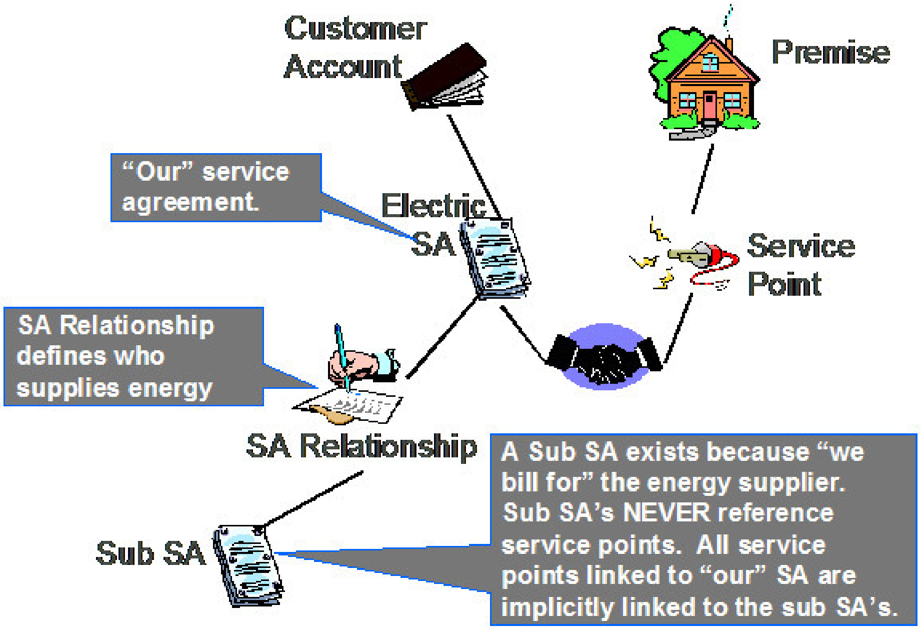
Consumption Relationships

When you set up a service provider, you must define your organization's consumption relationship with the service provider - a service provider may send the customers' consumption to you, you may send consumption to them, or you may have no consumption relationship with a given service provider.

The topics in this section provide a wealth of information about the various consumption relationships and the ramifications of each.

Only The Master SA Is Linked To Service Points

The following diagram makes the point that service points cannot be linked to sub SAs (i.e., service agreements that exist to hold charges associated with We Bill For Them service providers). This is because all service points associated with our "master" SA are implicitly linked to all sub SAs.


Customer Account
Premise
"Our" service agreement.
Electric SA
Service Point
SA Relationship defines who supplies energy
SA Relationship
A Sub SA exists because "we bill for" the energy supplier. Sub SA's NEVER reference service points. All service points linked to "our" SA are implicitly linked to the sub SA's.
Sub SA

It's important to understand why the system does not allow sub SAs to reference service points:

- The consumption associated with all service providers should be the same, otherwise the customer will receive inconsistent bills from different service providers.
- The easiest way to ensure consumption is the same for all service providers is to make sure that they all have the same service points (which they must if sub SAs "inherit" their service points from their master).

We Can Send Billed Consumption To Any Service Provider

You can send consumption to any service provider. It doesn't matter what their billing relationship is. Information on the service provider object tells the system if AND how to send consumption to a service provider.

Rather than send raw reads to service providers, we download consumption that has been calculated and snapshot onto the "master" bill segment. We send this billed consumption because:

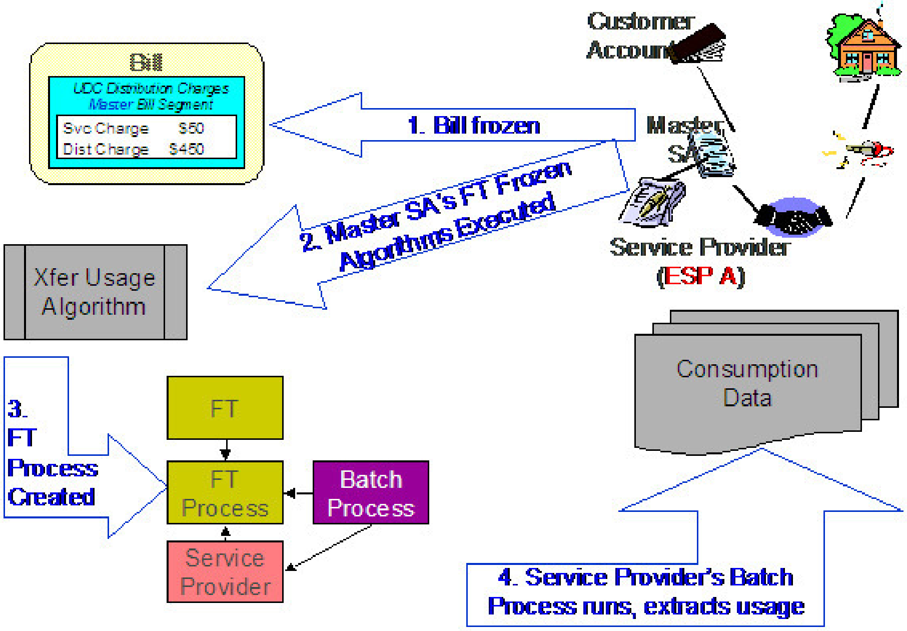
- It is clean and validated
- Register indexes have been subtracted
- Multiple registers have been summed

➤ **NOTE:**
We only send consumption after a bill segment is Frozen. If your organization supports We Bill For Them - Bill Ready service providers AND you are the source of consumption used by these service providers to calculate their charges, please refer to The Bill Ready Calculation Method for an explanation of how the system waits a given amount of time for the recipient of the consumption to interface their charges back to the system before sending the bill to the customer. In other words, the bill sent to the customer should contain the bill segment that triggered the consumption download as well as bill segments containing uploaded billable charges.

Routing Consumption To Service Providers - Technical Implementation

WARNING:
This section describes, technically, how we send consumption to service providers. If you aren't technically inclined, skip this section.

The following illustration shows the logical steps involved with sending consumption to service providers.



The following points explain the steps:

- When a financial transaction (FT) is frozen, the system executes the FT Freeze algorithms defined on the SA(s) SA type.
- If you've set up the system properly (i.e., you've put the appropriate FT Freeze algorithm on the master SA's SA type), one of these algorithms will determine if there are service providers associated with the master SA who need consumption. If so, it will insert a row on the FT Process table.
- Rows on the FT process table are used as "triggers" for batch processes. In this case, the batch process that is triggered is the one that downloads billable consumption to the service provider. If multiple service providers need consumption, multiple rows will be inserted. The ID of the batch process that is referenced on the trigger comes from the Service Provider's Consumption Download Process.

We Can Receive Consumption From Service Providers

We can receive consumption from any source. The system can receive consumption from service providers. Each customer must have:

- A premise
- A service point

- A device with measuring components sufficient to hold the interfaced consumption

NOTE:
You don't have to perfectly model the service points and meters. If you are not the distribution company, you may be worried about how to keep devices and service point information up-to-date. Be aware that you don't have to model the SPs and devices perfectly. Why? Because you just need to set up enough information so that consumption can be received accurately.

MDMAs And Service Cycles

WARNING:
This section uses terminology and concepts described in The Cyclical Meter Read Process.

A Meter Data Management Agency (MDMA) is a service provider who reads meters. In some locales, a meter can be read by a variety of MDMAs, in other locales there is no MDMA as the meter is read by the distribution company. If your organization has MDMAs, then you need to be aware of the following:

- MDMAs must be linked to service agreements as a service provider. Like all service providers, they can change over time.
- When an MDMA is reading a service agreement's meters, the MDMA may override the customer's service cycle (and schedule) with their own cycle. If they do this, the customer's service cycle in the MDMA's system is defined on the service agreement's SA relationship information for the MDMA. Note: whether or not a service provider can override a service cycle is controlled by a switch on the service provider's record. Refer to Service Provider - Main for more information.
- A service provider's service cycle schedules are maintained using the system's normal service cycle schedule. The ID of the service provider associated with each schedule is defined on the service cycle; in other words, if a service provider reads meters and they can override the customer's service cycle, the service provider's service cycles must be defined in the system. Refer to Designing Service Cycles, Routes, And Schedules for more information.
- At billing time, the system determines if a service agreement is covered by an MDMA. If so, it uses the service cycle defined on the service agreement's MDMA SA relationship record.
- If your organization ever reads the customers' meters, your regular read cycle should be maintained on your service points. You can think of the service cycle that is defined on a service agreement's MDMA SA relationship record as an override of your service points normal service cycle.
- If the meter read download process detects that a service point is linked to a service agreement with an active MDMA, it still creates a meter read download staging record; however, it marks it as Do Not Need To Read. This means that when a service point is no longer read by an MDMA, the meter read will be downloaded normally.
- Refer to We Can Receive Consumption From Service Providers for a description of how a service provider interfaces consumption into the system.

Deposits Issues

WARNING:
This section uses terminology and concepts described in The Big Picture Of Deposits.

Deposits should be held using normal deposit service agreements (SAs). You should NOT use the Sub Service Agreements (sub SA) functionality to hold or bill for deposits because deposit service agreements do not have the same state transition as do master SAs (e.g., you can activate or stop a deposit independent from its master).

However sub SAs can be covered by a deposit. If so, their SA type must reference a deposit class. To make the point, let's examine a few scenarios:

- Assume you have a sub SA for your own charges (this can happen when we use sub SAs to unbundle charges from the Master SA). In this case, it is likely that the sub SA and master SA will be in the same deposit class. This means that a single deposit SA would cover both the master and the sub SA.
- Assume They Bill For Us (Bill or Rate Ready). In this situation, we still have a master SA for our charges and we transfer the charges to the service provider who does the billing. In this case, it is likely that we would be holding the deposit for the service provider, not on the end-use customer. If we are in a situation where 1) we cannot hold a deposit against the service provider, and/or 2) we are not assured of the service provider paying us when the customer doesn't pay them, then we might want to put the master SA in a deposit class and hold a deposit against the customer's account using a normal deposit SA. We would not expect the service provider to bill the customer for the deposit, so we don't need a sub SA. We bill the customer directly for the deposit using our normal deposit SA.
- Assume We Bill For Them (Bill or Rate Ready). In this situation, we could hold a normal deposit SA for the customer's master SA. For sub SAs, we have two scenarios:
  - **We pay at billing time.** Since we purchase the receivable, we would want to increase our normal deposit to cover the Sub SA. To do this, the sub SA's SA type's deposit class should be the same as our master SA's deposit class.
  - **We pay at payment time.** It seems unlikely that we would want to hold a deposit on behalf of a service provider when we don't purchase the receivable. However, it is possible to do so by putting the sub SA into its own deposit class. If you did this, the system will require a separate deposit SA for the service provider's deposit. The system would calculate and refund such deposits using the algorithms defined on the new deposit SA's SA type's deposit class. It's important to be aware that the deposit is not held with respect to the specific service provider. Rather, it is just held in the system as separate deposit that could be used for any service agreement that belongs to its deposit class.

Credit and Collection Issues

WARNING:
This section uses terminology and concepts described in The Big Picture Of Credit & Collections (C&C).

C&C is only tricky if you deal with We Bill For Them service providers. We'll run through the service provider billing relationships to explain why:

- If we have a Dual relationship with a service provider, we don't have their debt, so we only have a responsibility to tell them when we cut a customer (via a Notification). We don't have to worry about collecting their debt.
- If we have a They Bill For Us relationship with a service provider, there is no debt on the customer's SA because it gets transferred to the service provider (and the service provider's SA will fall into arrears if they don't pay us).
- If we have a We Bill For Them or It's Us relationship with a service provider, the customer's debt associated with the service provider's service is maintained on a sub SA (i.e., it is segregated from our debt). This segregation of debt is both a powerful feature and a cause of administrative difficulties. The topics in this section provide more information about this issues.

Debt Class Recommendations

A service agreement's debt class is an important element in determining how a customer's debt is collected. In general, we recommend the following:

- If the service provider has a billing relationship of It's Us, we recommend the sub SAs belong to the same debt class as the "master". Why? Because both SAs' overdue debt should probably be grouped together under a single collection process.
- If you buy the receivable from the service provider (i.e., the service provider has a payment relationship of Pay at Billing Time), we recommend the sub SAs belong to the same debt class as the "master". Why? Because both SAs' overdue debt should probably be grouped together under a single collection process.
- If you don't buy the receivable from the service provider (i.e., the service provider has a payment relationship of Pay at Pay Time), you may want to use a different debt class on the sub SA. Why? Because you may collect the service provider's debt differently.

➤ **NOTE:**
Bottom line. If both the "master" and the sub SAs fall into arrears, you will have 1 or 2 collection processes, it all depends on the debt class assigned to each SA type.

● **FASTPATH:**
Refer to Automating Your C&C Activities for information describing how debt class plays a part in this processing.

Severing Service

Sub SAs and severance is tricky. Why?

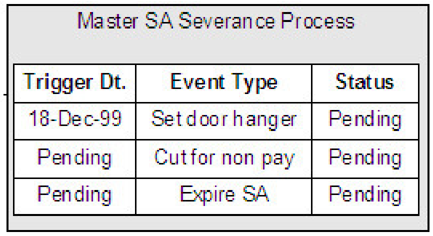
- Because it's possible for the master SA to be in arrears when the sub SA isn't (for all the standard reasons - directed payments, cancel / rebills, etc.).
- Because it's possible for the sub SA to be on one collection process and the master to be on another (due to different debt classes or different time lines).

Both of these situations could result in severance starting for only one of the service agreements in the master / sub relationship. However, YOU CAN'T CUT SERVICE FOR ONE WITHOUT CUTTING THE OTHER because there is only one service point.

Before we describe how to deal with this conundrum, we'd like to remind you that the system starts a unique severance process for each SA (sub or normal) to be severed. It only creates a severance process for those service agreements linked to a collection process when the collection process' Start Severance event is activated. The type of severance process that is created is controlled by each service agreement's SA Type's severance criteria. Please keep in mind the following when designing these severance processes:

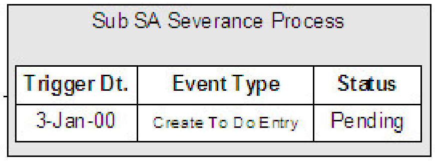
- Only The "Master" Service Agreement Is Linked To Service Points. This means only master SAs should have a "cut for non payment" severance event. Note: typically, such a severance process will expire the "master SA" several days after the cut event if funds are NOT received.
- As described under Sub SA State Transition, a sub SA becomes Pending Stop (and eventually Stopped) when its "master" is stopped. This means the sub SAs will be finaled when the master is finaled.
- If you start severance on a sub SA when the master isn't being severed, you have a problem because you can't cut the sub SA independent from the master SA.

We'll use an example to illustrate how you should design your severance processes to deal with the above challenge. Assume you have a master and a sub SA where both are being managed under the same collection process. Also assume that the Start Severance event kicks off on 18-Dec-1999. In this situation, we'd recommend the following severance processes to be kicked off.



Master SA Severance Process

Trigger Dt.	Event Type	Status
18-Dec-99	Set door hanger	Pending
Pending	Cut for non pay	Pending
Pending	Expire SA	Pending



Sub SA Severance Process

Trigger Dt.	Event Type	Status
3-Jan-00	Create To Do Entry	Pending

Notice that the sub SA's severance process contains a single event that generates a To Do Entry on a date in the future of the Expire SA event on the Master SA. This entry should be something like "sub being severed independent of its master". This event will only be triggered if the master SA is paid off and the sub SA isn't. Why? Because if the master SA's Expire SA event is executed, the Sub SA will be Stopped and stopping a SA cancels outstanding severance processes. If the sub gets paid, the system will cancel the sub's severance process.

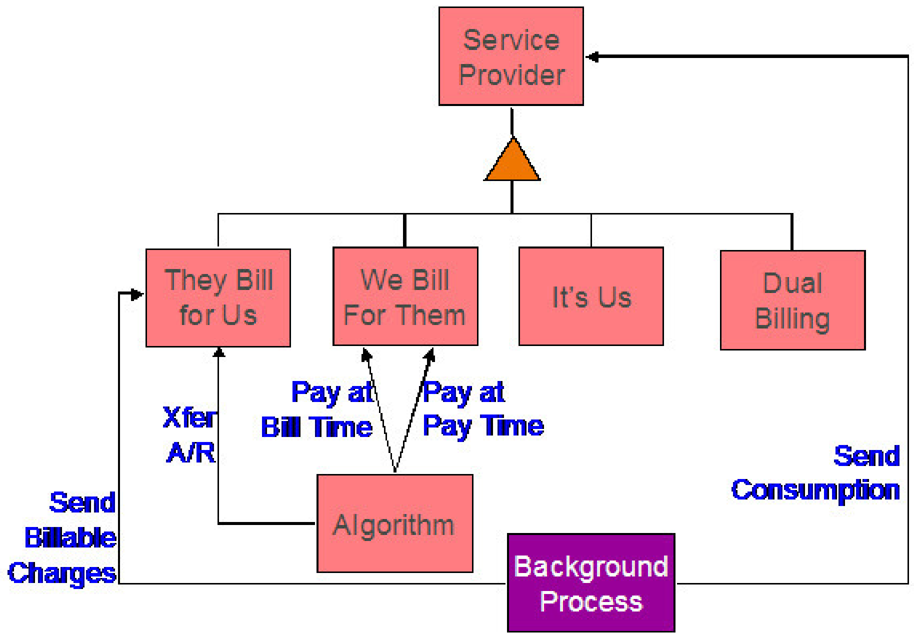
Let's change the example and assume that the master starts severance and the sub doesn't. In this situation, the master SA will eventually hit the Expire SA event and the sub SA will also stop. There's no alternative.

And let's change the example again and assume the sub starts severance and the master doesn't. In this situation, the To Do Entry will only be created X days after the start of severance. If you can't stand this date being X days in the future of the creation of the severance process, create an "Severance Criteria Algorithm" that checks if the master is not being severed or collected and generates a different severance process (with a different start date). Refer to Designing Your Severance Procedures for more information about Severance Criteria Algorithms.

An Object-Oriented Perspective Of Service Providers

WARNING:
Skip this section if you are not technically inclined.

The following object diagram illustrates in a concise format the various types of service providers and the plug in algorithms and processes available for each.



How Do You Communicate With Service Providers?

You communicate with service providers (and they communicate with you) using the following mechanisms:

- You can use traditional interface techniques. For example, if you send consumption to service providers every month (so they can compute their charges), you could use the Consumption Download interface.
- You can use "notifications" to communicate with service providers. Notifications are electronic transactions that service providers exchange to communicate information about a customer. For example, you could use a notification to send a message to an energy supplier when a customer stops service. Note: the term "Direct Access Service Request" (DASR) is synonymous with Notification.

The difference between notifications and traditional interfaces is subtle. Think of notifications as a generic interface that can be used to communicate many different things (e.g., you can use notifications to advise stops, meter exchanges, supplier switches). Traditional interfaces communicate only one thing (and you need one traditional interface for each "thing"). For example, one interface is devoted to downloading consumption, another is responsible for uploading pass through charges, etc.

Off Cycle Switch

If the customer care and billing application integrates with meter data management (mdm) system, it may be set up to generate sub-bill segments for each and every service provider having **We Bill for Them - Rate Ready** or **It's Us** billing relationship, that is effective in the billing period. This requires that the meter data management system can calculate the consumption for each effective SA relationship period.

If the switch is from or to a service provider having **They Bill for Us** billing relationship, separate bills have to be generated manually for each relationship effective in the billing period. This restriction is due to requirement for CCB application to generate a bill for a **They Bill for Us** relationship, so that the customer's charges are calculated and eventually transferred to the service provider. The generated bill for **They Bill for Us** billing relationship is not sent to the customer.

Billing Master Configuration

To enable Off Cycle Switch, navigate to **Admin** > **General** > **Master Configuration** and add a new entry for Billing Master Configuration.

Specify the values for **Off Cycle Switch Option**, **Implementation Date** and **Off Cycle Switch Option for Old Bill**. Refer to the embedded help on billing master configuration portal for details.

Rate Schedule

Rate schedules assigned to the sub-SA of service provider, having **We Bill for Them - Rate Ready** or **It's Us** billing relationship, must be configured with an SQ rule (for component-based engine) or a pre-processing calculation rule (for rule-base engine) that will obtain the sub-usage's consumption and load it into the bill segment SQ collection during rate application.

For more information, refer to the base package algorithm types provided for this purpose: US (component based engine) and C1-ACRSUBUSG (rule based engine).

Sub-SA Type

When implementing this feature, sub-SA types of service provider having **We Bill for Them - Rate Ready** or **It's Us** billing relationship must be configured as non-bill determinant.

Usage Request

Usage request is sent to the Oracle Utilities Meter Data Management (MDM) system, to obtain the calculated consumption for the billing period - this is used to calculate master bill segment's charges.

In addition, sub-usage request is sent to the mdm system, for each relationship that is effective during the billing period. Calculated consumption on these sub-usage requests is used to calculate sub-bill segment charges.

Sending Consumption to Service Providers

Calculated consumption on these sub-usage requests also serves as the source for consumption download process if consumption is required to be sent to the service provider. A base package algorithm type - C1-STGCNSMP, stages the download by creating an entry in the general process table, capturing the sub-usage id, batch control for consumption download and the batch number.

> **NOTE:** It is implementation responsibility to build the consumption download batch process that will extract the consumption for sub-usage request as required.

Designing Your SA Relationship Types and Service Providers

As explain in A Service Agreement Can Have Many Types Of Relationships, SA relationship types and service providers are only required when you subcategorize your service agreements.

The topics in this section describe how to design your SA relationship types.

Designing SA Relationship Types

The easiest way to design SA relationship types is to start with the matrix of SA types designed in SA Types And The Financial Design. For each SA type in the matrix, determine if either of the following questions is true:

- Can companies other than your own provide some subcategory of the service (and do you have some type of interaction with these companies)? For example, this would be true if you are an energy supply company because a different company is responsible for distributing the power to the customer (and you probably exchange consumption and financial transactions with this company).
- Does your organization use different rates for subcategories of the service? For example, this would be true if you use separate rates for water and wastewater service (even though both are based on the customer's water consumption).

If either of the above is true, you will need a SA relationship type for each subcategory of service.

We'll use an example to help make the point. Using the SA types we designed earlier (see SA Types And The Financial Design), we'll assume the following:

- We are designing the system for an electric, gas, water, waste water, and cable utility.
- Residential electric customers have a choice of energy supplier.
- Commercial and industrial electric customers can choose an energy supplier and a meter service provider.
- Commercial and industrial gas customers have a choice of energy supplier.

➤ **NOTE:**
We are not showing most of the SA types that we designed earlier because they do not have subcategories of service.

CIS BU/ SA Type	SA Relationship Type
G/COM	Energy supply
G/IND	Energy supply
E/RES	Energy supply
E/COM	Energy supply
	Meter service
E/IND	Energy supply
	Meter service

➤ **NOTE:**
Notice that we did not design a SA relationship type for our own distribution service. This is because our relationship type is implied (e.g., if you are a distribution company, you do not have to set up a SA

relationship type for distribution service because the customer's "master" service agreement is implicitly associated with distribution service).

Designing Service Providers

After you design your SA relationship types, you need to list every potential service provider for each SA relationship type.

CIS BU/ SA Type	SA Relationship Type	Service Provider
G/COM	Energy supply	AmeriGas
		TransGas
		Green Power
G/IND	Energy supply	AmeriGas
		TransGas
		Green Power
E/RES	Energy supply	Green Power
		Cheap Power
		Us
E/COM	Energy supply	ElectriCorp
		Cheap Power
		TeniCorp
	Meter service	MeterCorp
		Us
E/IND	Energy supply	ElectriCorp
		Cheap Power
		TeniCorp
	Meter service	MeterCorp
		Us

Next, list each unique service provider identified above:

Service Provider
AmeriGas
TransGas
Green Power
Cheap Power
Us
ElectriCorp
TeniCorp

You will have at least one service provider for each entry in the above list. However, you may have to set up more than one service provider in the system for a given company. The topics below explain how this happens.

Billing Relationship Segmentation

As described under Billing Relationships, a service provider may bill for you, you may bill for them, or you may each send a separate bill to the customer. In the table below, we have shown the assumed billing relationships for each service provider.

Service Provider	Billing Relationship
AmeriGas	We Bill For Them - Bill Ready
TransGas	We Bill For Them - Bill Ready
Green Power	We Bill For Them - Rate Ready
Cheap Power - Res	We Bill For Them - Rate Ready
Cheap Power - Com/Ind	Dual Billing
Us - Billable	It's Us
Us - Non billable	None
ElectriCorp	They Bill For Us - Bill Ready
TeniCorp	We Bill For Them - Bill Ready
MeterCorp	Dual Billing

Notice that we had to introduce additional service providers:

- Cheap Power has two service providers - one for residential customers, the other for commercial/industrial customers. This is necessary because we provide billing service for them for residential customers (We Bill For Them - Rate Ready), but for commercial and industrial customers they bill for themselves (Dual Billing).
- Our own service provider (the service provider known as "Us") has two service providers - one for energy supply because we bill for the energy we supply (It's Us), and another for meter service because we don't create bills for meter service (None).

We Bill For Them - Payment Relationship Segmentation

As described under Pay At Bill Time vs. Pay At Pay Time when you provide billing service for a service provider you have to define if you pay the service provider when you bill the customer OR only later, when the customer pays you.

In the table below, we have shown the payment relationships for each service provider.

Service Provider	Billing Relationship	Payment Relationship
AmeriGas	We Bill For Them - Bill Ready	Pay At Billing Time
TransGas	We Bill For Them - Bill Ready	Pay At Billing Time
Green Power	We Bill For Them - Rate Ready	Pay At Billing Time
Cheap Power - Res	We Bill For Them - Rate Ready	Pay At Billing Time
Cheap Power - Com/Ind	Dual Billing	N/A

Us - Billable	It's Us	N/A
Us - Non billable	None	N/A
ElectriCorp	They Bill For Us - Bill Ready	N/A
TeniCorp - Com	We Bill For Them - Bill Ready	Pay At Billing Time
TeniCorp - Ind	We Bill For Them - Bill Ready	Pay At Pay Time
MeterCorp	Dual Billing	N/A

Notice that we had to introduce an additional service provider for TeniCorp because for commercial customer we purchase the receivable (Pay At Bill Time), but for industrial customers we only pay them when we're paid by the customer (Pay At Pay Time).

Consumption Relationship Segmentation

As described under Consumption Relationships a service provider may send the customers' consumption to you, you may send consumption to them, or you may have no consumption relationship with a given service provider.

In the table below, we have shown the consumption relationships for each service provider.

Service Provider	Billing Relationship	Payment Relationship	Consumption Relationship
AmeriGas	We Bill For Them - Bill Ready	Pay At Billing Time	We Send Consumption
TransGas	We Bill For Them - Bill Ready	Pay At Billing Time	We Send Consumption
Green Power	We Bill For Them - Rate Ready	Pay At Billing Time	N/A
Cheap Power - Res	We Bill For Them - Rate Ready	Pay At Billing Time	N/A
Cheap Power - Com/Ind	Dual Billing	N/A	N/A
Us - Billable	It's Us	N/A	N/A
Us - Non billable	None	N/A	N/A
ElectriCorp	They Bill For Us - Bill Ready	N/A	N/A
TeniCorp - Com	We Bill For Them - Bill Ready	Pay At Billing Time	We Send Consumption
TeniCorp - Ind	We Bill For Them - Bill Ready	Pay At Pay Time	We Send Consumption
MeterCorp	Dual Billing	N/A	N/A

Notice that we didn't have to proliferate service providers due to consumption relationships.

Other Segmentations

The earlier parts of this discussion described the most common factors that cause the creation of service providers. However, many obscure factors could cause the introduction of more service providers. In this section we explain these more obscure factors.

Payment Method

As described under Pay At Bill Time vs. Pay At Pay Time, when you provide billing service for a service provider you have to pay the service provider at some point in time. The algorithm that defines the amount to pay (and how the related adjustment is generated) is defined on the service provider record. If a service provider has different payment algorithms for different customer segments, you must split the service provider accordingly.

Transfer Receivable Method

As described under When They Bill For Us, They Owe Us Money, a service provider will owe you money if they provide billing service for you. The algorithm that defines how to transfer the customer's receivable to the service provider is defined on the service provider record. If a service provider has different transfer A/R algorithms for different customer segments, you must split the service provider accordingly.

Billable Charge Download Process

As described under They Bill For Us - Bill Ready, billable charges are interfaced to service providers who provide billing service for you. The background process that performs the interface of billable charges is defined on the service provider record. If a service provider has different billable charge interfaces for different customer segments, you must split the service provider accordingly.

Consumption Download Process

As described under We Can Send Billed Consumption To Any Service Provider, consumption can be sent to any service provider. The background process that performs the interface of consumption is defined on the service provider record. If a service provider has different consumption interfaces, you must split the service provider accordingly.

Notification Upload Processing

A service provider can send you notifications. Whenever a notification is uploaded, the system creates a workflow process to process each such notification. The type of workflow process that's created is controlled by the service provider's workflow process profile. If a service provider requires a different workflow process for a given type of notification (for whatever reason), you must split the service provider accordingly.

Notification Download Processing

The system will send notifications to service providers when something noteworthy happens and when information is needed from a service provider. The type of notification that is sent to a service provider and the background process that interfaces the notification to the service provider is defined on the service provider's notification download profile. If a service provider requires a different type of notification to be sent or they have different interface protocols for a given type of notification, you must split the service provider accordingly.

Financial Settlement Service Agreement

As described under Service Providers Have Service Agreements Too, service providers have service agreements. These service agreements contain how much you owe the service provider (if you bill for them) and how much they owe you (if they bill for you). If you want to have separate service agreements for financial settlements associated with distinct customer segments, you must split the service provider accordingly.

Geographic Area Segmentation

The following table reflects the new service providers that were added since we started Designing Service Providers.

CIS BU/ SA Type	SA Relationship Type	Service Provider
G/COM	Energy supply	AmeriGas
		TransGas
		Green Power
G/IND	Energy supply	AmeriGas
		TransGas
		Green Power
E/RES	Energy supply	Green Power
		Cheap Power - Res
		Us - Billable
E/COM	Energy supply	ElectriCorp
		Cheap Power - Com/Ind
		TeniCorp - Com
	Meter service	MeterCorp
		Us - Non billable
E/IND	Energy supply	ElectriCorp
		Cheap Power - Com/Ind
		TeniCorp - Ind
	Meter service	MeterCorp
		Us - Non billable

Next, determine the postal code ranges in which a service provider is allowed to provide service.

CIS BU/ SA Type	SA Relationship Type	Service Provider	Postal Range
G/COM	Energy supply	AmeriGas	94000 - 95999
		TransGas	94000 - 95999
		Green Power	94000 - 95999
G/IND	Energy supply	AmeriGas	94000 - 95999
		TransGas	94000 - 95999
		Green Power	94000 - 95999
E/RES	Energy supply	Green Power	94000 - 95999
			93000 - 93999

		Cheap Power - Res	94000 - 95999
		Us - Billable	94000 - 95999
E/COM	Energy supply	ElectriCorp	94000 - 95999
		Cheap Power - Com/Ind	94000 - 95999
		TeniCorp - Com	94000 - 95999
	Meter service	MeterCorp	94000 - 95999
		Us - Non billable	94000 - 95999
E/IND	Energy supply	ElectriCorp	94000 - 95999
		Cheap Power - Com/Ind	94000 - 95999
		TeniCorp - Ind	94000 - 95999
	Meter service	MeterCorp	94000 - 95999
		Us - Non billable	94000 - 95999

Next, we need to strip off the SA types because the postal ranges are defined for combinations of service provider and SA relationship type. Notice the problem - we have a service provider - Green Power has different postal ranges for the same SA relationship type. You have two ways to fix this problem, you can split your service provider (have one for the gas and another for the electric), or you can split the SA relationship type (have one for the gas and another for the electric). We've chosen the former in our example.

SA Relationship Type	Service Provider	Postal Range
Energy supply	AmeriGas	94000 - 95999
	TransGas	94000 - 95999
	Green Power - Gas	94000 - 95999
Energy supply	AmeriGas	94000 - 95999
	TransGas	94000 - 95999
	Green Power - Gas	94000 - 95999
Energy supply	Green Power - Electric	94000 - 95999
		93000 - 93999
	Cheap Power - Res	94000 - 95999
	Us - Billable	94000 - 95999
Energy supply	ElectriCorp	94000 - 95999
	Cheap Power - Com/Ind	94000 - 95999
	TeniCorp - Com	94000 - 95999
Meter service	MeterCorp	94000 - 95999
	Us - Non billable	94000 - 95999
Energy supply	ElectriCorp	94000 - 95999
	Cheap Power - Com/Ind	94000 - 95999
	TeniCorp - Ind	94000 - 95999
Meter service	MeterCorp	94000 - 95999
	Us - Non billable	94000 - 95999

In the table below, we have shown the final list of service providers.

Service Provider	Billing Relationship	Payment Relationship	Consumption Relationship
AmeriGas	We Bill For Them - Bill Ready	Pay At Billing Time	We Send Consumption
TransGas	We Bill For Them - Bill Ready	Pay At Billing Time	We Send Consumption
Green Power - Electric	We Bill For Them - Rate Ready	Pay At Billing Time	N/A
Green Power - Gas	We Bill For Them - Rate Ready	Pay At Billing Time	N/A
Cheap Power - Res	We Bill For Them - Rate Ready	Pay At Billing Time	N/A
Cheap Power - Com/Ind	Dual Billing	N/A	N/A
Us - Billable	It's Us	N/A	N/A
Us - Non billable	None	N/A	N/A
ElectriCorp	They Bill For Us - Bill Ready	N/A	N/A
TeniCorp - Com	We Bill For Them - Bill Ready	Pay At Billing Time	We Send Consumption
TeniCorp - Ind	We Bill For Them - Bill Ready	Pay At Pay Time	We Send Consumption
MeterCorp	Dual Billing	N/A	N/A

Designing Your SA Types And Start Options For Sub SAs

When you were Designing Service Providers, you defined the service providers that were valid for every combination of SA type and SA relationship type. If you provide billing services for another service provider or if you subcategorize your own services, you another task - you have to design the SA types for your sub SAs.

As described earlier, there will be a separate Sub Service Agreement for every SA relationship for which we calculate a bill segment. Every sub SA must reference an SA type. The following table shows sample SA types (notice that they are only used for We Bill For Them and It's Us service providers).

CIS BU/ SA Type	SA Relationship Type	Service Provider	SA Type(s) for Sub SAs
G/COM	Energy supply	AmeriGas	AG1
		TransGas	TG1
		Green Power - Gas	GP-GC1
G/IND	Energy supply	AmeriGas	AG1
		TransGas	TG1
		Green Power - Gas	GP-GI1
E/RES	Energy supply	Green Power - Electric	GP-ER1
		Cheap Power - Res	CP-ER1
		Us - Billable	US-ER1
E/COM	Energy supply	ElectriCorp	Not applicable - they bill for us
		Cheap Power - Com/Ind	Not applicable - dual billing
		TeniCorp - Com	TC1
	Meter service	MeterCorp	Not applicable - dual billing
		Us - Non billable	Not applicable - no billing

E/IND	Energy supply	ElectriCorp	Not applicable - they bill for us
		Cheap Power - Com/Ind	Not applicable - dual billing
		TeniCorp - Ind	TC1
	Meter service	MeterCorp	Not applicable - dual billing
		Us - Non billable	Not applicable - no billing

The design steps required to set up these SA types are similar to those described under Designing SA Types For Service Agreements With Service Points. The following points provide a few suggestions that will help you design your SA types for sub SAs:

- The business unit should be the same as defined for the master service agreement.
- Service type should be set up as per Service Segmentation.
- Distribution code should be set up as per Receivable Segmentation.
- Obviously, the sub SA switch should be turned on.
- Start options:
    - If the SA type is used for a We Bill For Them - Rate Ready service provider or for yourself, the start options should be Required because rates, contract riders and/or contract values will be populated on the sub SA from a start option. Refer to Automatic Creation of Sub SAs for more information.
    - If the SA type is used for a We Bill For Them - Bill Ready service provider, the start options should be not allowed because we don't need to default rates, contract riders and/or contract values on billable charge sub SAs.
- The payment distribution and late payment information should be set up as for any SA type. Refer to Cash Distribution Segmentation and Late Payment Charge Segmentation for more information.
- If the SA type is used for We Bill For Them - Bill Ready service provider, the special role should be Billable Charge, otherwise it should not be used. Remember, you should not use sub SAs for Cash Deposits.
- Deposit class should be used if the sub SA is covered by a deposit. Refer to Deposits Issues for more information.
- The one time charge switch should be off.
- Bill segment type:
    - If the SA type is used for a We Bill For Them - Bill Ready service provider, the bill segment type should reference a bill segment creation algorithm that creates bill segments from billable charges (and generates bill segment errors until the last night of the bill cycle). Refer to The Bill Ready Calculation Method for more information.
    - If the SA type is used for a We Bill For Them - Rate Ready service provider, the bill segment type should reference a get consumption algorithm that gets consumption from the master SA, and a bill segment creation algorithm that applies a rate.
- Specify a characteristic premise is required if the sub SA is associated with premise-oriented service.
- The calendar billing options should not be used.
- The recurring charge information should not be used.
- Sub SAs used for We Bill For Them - Rate Ready service providers should NOT be eligible for budgets and therefore the eligible for budget switch should be off. This admonition is given because budget billing causes current amount due to be out-of-sync with payoff amount due and we don't want this to happen for sub SAs. Why? Because we use the bill segments associated with these sub SAs to determine how much we owe the service provider.
- If the SA type is used for a We Bill For Them - Rate Ready service provider or for your own company, rates need to be set up; otherwise they should not be used.
- Sub SAs never reference service points. Refer to Only The "Master" Service Agreement Is Linked To Service Points for more information.

- Adjustment types profiles should be set up accordingly.
- Refer to Credit and Collection Issues for recommendations in respect of the debt class, write off debt class, and severance criteria associated with these SA types.
- Billable charge templates should not be used.
- Completion algorithms cannot be used for sub SAs.
- If the SA type is used for a We Bill For Them service provider, you should link to the SA type the FT freeze algorithm that controls how we pay the service provider. Refer to Pay At Bill Time vs. Pay At Pay Time for how this algorithm is used.
- The creation of bill segments for the sub SAs occurs after the bill segment for the related master SA is created. If you populate a billing processing sequence on an SA type for a sub SA, it is used to control the order in which the sub SAs for a given master SA are processed relative to each other.

Refer to Setting Up SA Types for how to set up these new SA types in the system.

Reference Send Consumption Algorithm On Master SA Types

As explained under We Can Send Billed Consumption To Any Service Provider, when a master SA's bill segment is frozen, the system must check if there are any service providers who need the bill segment's consumption. If so, it sets up the data necessary to interface the master SA's consumption (snapshot on the bill segment) to the service provider(s). The system will only do this if you specify an appropriate FT Freeze Algorithm on the master SA types. Refer to SA Type - Algorithm (FT Freeze Algorithm) for more information.

Reference TBFU Algorithm On Master SA Types

As explained under They Bill For Us, when a bill is completed, the system needs to check if there are any service providers who bill for us associated with the bills "master" SAs. If so, it sets up the data necessary to interface the master SA's charges to the service provider and to transfer the receivable balance from the customer to the service provider. The system will only do this if you specify an appropriate FT Completion Algorithm on the master SA types. Refer to SA Type - Algorithm (Bill Completion Algorithm) for more information.

➤ **NOTE:**
If there are multiple master SAs on a bill, the financial transactions associated with each respective master SA could be routed to different service providers (e.g., one service provider could bill for gas and another could bill for electricity). Refer to Different Service Providers Can Bill Different Services for more information.

Designing SA Types For Service Provider Financial Settlements

As explained in Service Providers Have Service Agreements Too, We Bill For Them and They Bill For Us service providers require a service agreement. You must create SA types for these types of service agreements. The following points provide a few suggestions that will help you design these financial settlement SA types:

- Service type should probably be a non-service oriented service type.

- Distribution code for We Bill For Them settlement SAs should be a payable account (or treat it as a "contra" receivable. Refer to Receivable Segmentation for more information.
- The sub SA switch should be turned off.
- Start options are not allowed.
- The payment distribution and late payment information should be set up as for any SA type. Refer to Cash Distribution Segmentation and Late Payment Charge Segmentation for more information.
- Special role should not be used.
- Deposit class should be used if the settlement service agreement is covered by a deposit. This would probably only be used for They Bill For Us service providers (because they will owe us money).
- The one time charge switch should be off.
- These service agreements are not billable and therefore none of the billing information should be specified.
- The characteristic premise switch should be off.
- Rates should not be used.
- Service points should not be used.
- Adjustment types profiles should be set up accordingly.
- Debt class, write off debt class, and severance criteria should be set up accordingly.
- Billable charge templates should not be used.
- Completion algorithms should not be used.
- Freeze algorithms should not be used.

Refer to Setting Up SA Types for how to set up these new SA types in the system.

Setting Up SA Relationship Information

In the previous section, Designing Your SA Relationship Types and Service Providers, we presented a case study that illustrated a mythical organization's SA relationship information. In this section we explain how to set up this information.

Setting Up SA Relationship Types

Open **Admin** > **Open Market** > **SA Relationship Type** to define your SA relationship types. Refer to Designing SA Relationship Types for more information.

Description of Page

Enter an **SA Relationship Type** code and **Description** for every relationship type.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference CI_SA_REL_TYPE.

Setting Up Service Providers

The topics in this section describe how to set up service providers.

Service Provider - Main

Open **Admin** > **Open Market** > **Service Provider** > **Add** to define core information about a service provider.

Description of Page

Enter a unique **Service Provider** code for the service provider.

End a brief **Description** of the service provider.

If you communicate with this service provider via notification messages or outbound messages, indicate the service provider's **External System**.

If you send notifications to this service provider, select a **Notification DL** (download) **Profile** that is used to define the configuration of the outgoing messages.

Select the **Person ID** that contains this service provider's phone numbers and demographic information.

If you bill for the service provider or if they bill for you, select the **Service Agreement** that holds how much you owe them or they owe you. Refer to Service Providers Have Service Agreements Too for more information.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference CI_SPR.

Service Provider - Detail

Open **Admin** > **Open Market** > **Service Provider** > **Search** and navigate to the **Detail** page to define additional information about your service providers.

Description of Page

Define the **Billing Relationship** you have with the service provider. Refer to Billing Relationships for more information.

If you provide billing services for this service provider (i.e., the **Billing Relationship** is We Bill For Them), define the **Payment Relationship**. Refer to Pay At Bill Time vs. Pay At Pay Time for more information. You may not be paying some service providers as such. Rather, the customer's receivables are simply transferred to the service provider, e.g., when you calculate discounts for special negotiated terms. For these service providers, choose a payment relationship of Pay SPr Not Applicable.

Define the **Consumption Relationship** you have with the service provider. Refer to Consumption Relationships for more information.

If a service provider reads meters and they can override the customer's service cycle, turn on **Overrides ServiceCycle**. Refer to MDMAs And Service Cycles for more information.

As described under Pay At Bill Time vs. Pay At Pay Time, when you provide billing service for a service provider you have to pay the service provider at some point in time. The **Pay Service Provider Algorithm** defines the amount to pay and how the related adjustment is generated. Refer to Paying The Service Provider - Technical Implementation for more information about how this algorithm is used. If you haven't done so already, you must set up this algorithm in the system. To do this:

- Create a new algorithm (refer to Setting Up Algorithms).
- On this algorithm, reference an Algorithm Type that increases how much is owed the service provider. Click here to see the algorithm types available for this plug-in spot.

As described under When They Bill For Us, They Owe Us Money, a service provider will owe you money if they provide billing service for you. The **Transfer A/R Algorithm** defines how to transfer the customer's receivable to the service provider. Refer to A/R Transfer - Technical Implementation for more information about how this algorithm is used. This algorithm is also used to transfer receivables when you calculate discounts for special negotiated terms. If you haven't done so already, you must set up this algorithm in the system. To do this:

- Create a new algorithm (refer to Setting Up Algorithms).
- On this algorithm, reference an Algorithm Type that transfers financial transactions from the customer to the service provider. Click here to see the algorithm types available for this system event.

As described under They Bill For Us - Bill Ready, billable charges are interfaced to service providers who provide billing service for you. The **Billable Charge Download Process** performs the interface of billable charges to the service provider. Refer to Routing Billable Charges To Service Providers - Technical Implementation for more information about how this process is used.

As described under We Can Send Billed Consumption To Any Service Provider, consumption can be sent to any service provider. The **Consumption Download Process** performs the interface of consumption to the service provider. Refer to Routing Consumption To Service Providers - Technical Implementation for more information about how this process is used.

Service Provider - SA Relationship Type

Open **Admin** > **Open Market** > **Service Provider** > **Search** and navigate to the **SA Relationship Type** page to define the types of relationships (e.g., energy supplier, energy distributor, meter data management agency) associated with a service provider and the postal code ranges in which the service provider operates.

Description of Page

Use the **SA Relationship Types** collection to define this service provider's **SA Relationship Types**. Use the collection that appears in the grid to define the **Postal Code** ranges in which this service provider is allowed to operate for each **SA Relationship Type**.

⚠ **IMPORTANT:**
After defining the SA relationship types that can be associated with a service provider, you must then define the SA types on which the service provider / SA relationship type combination can be used. This information is defined using Setting Up SA Relationships For SA Types.

Service Provider - Bill Messages

Open **Admin** > **Open Market** > **Service Provider** > **Search** and navigate to the **Bill Messages** page to define bill messages to appear on bills that contain charges associated with a service provider.

Description of Page

Use the **Bill Messages** collection to define **Bill Message** codes that should appear on bills that contain charges associated with a given service provider. For each message, also specify the **Start Date** and **End Date** when such a message should appear on the bill (leave **End Date** blank if the message should appear indefinitely).

Where Used

The system snaps bill messages on a bill during bill completion. Refer to The Source Of Bill Messages for more information.

Setting Up SA Types and Start Options For Sub SAs

The SA types and start options described under Designing Your SA Types And Start Options For Sub SAs must be set up. Refer to Setting Up SA Types for how to do this.

Setting Up SA Types For Financial Settlements

The SA types described under Designing SA Types For Service Provider Financial Settlements must be set up. Refer to Setting Up SA Types for how to do this.

Update Master SA Types With FT Freeze and Bill Completion Algorithms

Refer to SA Type - Algorithm for more information.

Setting Up SA Relationships For SA Types

SA Type SA Relationship Type - Main

Open **Admin** > **Open Market** > **SA Type SA Rel. Type** > **Add** to define the types of SA relationships and service providers than can be associated with a SA type.

Description of Page

Define the **SA Relationship Type** that can be associated with service agreements of this **SA Type**.

➤ **NOTE:**
You may only define SA Relationship Types for "master service agreements".

Turn on the **Required** switch if this **SA Relationship Type** must be defined for service agreements of this type. Refer to Defaulting Relationship Types And Defaulting Service Providers and Required Relationship Types and Billing for more information.

Indicate if **Gaps in SA Relationships** of this type that are associated with this service agreement type are Allowed or Not Allowed. You should only select Allowed if relationships of this type can be expired without a relationship with another service provider to replace it. Deregulated relationships typically should not have gaps in the relationship. For example, a relationship with an energy service provider should not expire unless a relationship with another energy service provider replaces it. Refer to negotiated terms for an example of SA relationship types that allow gaps in SA relationships.

Use the collection to define the **Service Provider**s who can be associated with this **SA Relationship Type** on service agreements of this **SA Type**.

➤ NOTE:

Only **Service Provider**s previously defined as being valid for the **SA Relationship Type** can be specified (refer to Service Provider - SA Relationship Type for how to link a service provider to a SA relationship type).

Turn on **Default SPR** if the **Service Provider** should be defaulted on newly created SA relationships. Refer to Defaulting Relationship Types And Defaulting Service Providers for more information.

Use **Status** to control if the **Service Provider**'s relationship to the SA Type / SA Relationship type is Active or Inactive. Only Active service providers can be linked to service agreements of this type. The system allows Inactive service providers in order to support historical service providers who are no longer active and to allow you to set up new service providers in advance of their start date.

Use the drill down button adjacent to a service provider to view the valid Sub SA Types. Alternatively, navigate the Sub SA Type tab and scroll until you find the desired service provider.

Where Used

When a new SA relationship is defined for a service agreement, the system uses this information to make sure the relationship is valid and that the associated service provider is valid.

SA Type SA Relationship Type - Sub SA Type

Open **Admin** > **Customer** > **SA Type SA Rel Type** > **Search** and navigate to the **Sub SA Type** page to define valid sub SA types for service providers associated with a SA type.

If you provide billing services for the service provider (i.e., the service provider's billing relationship is We Bill For Them) or if you subcategorize your own charges (i.e., the service provider is your organization and it has a billing relationship of It's Us) a Sub Service Agreement will be created for the service provider.

Description of Page

The information in the collection defines the valid **Sub CIS Division** and **Sub SA Type**s of these sub service agreements. Those entries marked as **Create Initially** are used by the process that creates sub SAs for new SA relationships. This background process uses this information as follows:

- If the sub SA's SA type doesn't use Start Options (as defined on SA Type - Main), the background process simply creates a sub SA with the given SA type. Note: these types of sub SAs are typically used for service providers who send their charges to you (i.e., they have a billing relationship of We Bill For Them - Bill Ready). This is because Billable Charge service agreements are used for these types of service agreements and billable charge service agreements contain very little information.
- If the sub SA's SA type uses Start Options (as defined on SA Type - Main), the **Start Option** defined in the collection is used to populate the sub SA with default values (e.g., rate, contract rider, etc.). Refer to Setting Up Start Options for more information.

Where Used

The process that creates sub SAs for new SA relationships uses this information to determine the number and type of sub SAs to create for each SA relationship.

Negotiated Terms

The topics in this section describe the use of SA relationship functionality for applying certain types of negotiated terms. It assumes that you are familiar with SA relationship functionality in general.

> **NOTE:**
> **Negotiated terms are optional.** The functionality described in this section is only relevant if your organization offers this functionality.
>
> **Umbrella agreements**. If the Contract Management module is not turned off, you may also choose to use umbrella agreements to manage the functionality described here.

Imagine that the head office for a multi-site organization negotiates special terms that cover a number of its sites, each of whom have their own account and service agreement. The negotiated terms typically involve discounts. These discounts may be realized

- Under a single service agreement, separate and distinct from the service agreements that are covered, or
- Individually, for each covered service agreement. In this case, two sub-scenarios exist:
  - The discounts may be reflected on the bill segment of each service agreement, or
  - The discounts may be transferred to another service agreement (the group's service agreement). The discount does NOT appear on the individual bill segment for the service agreements that are covered by the negotiated terms.

SA relationships track and manage complex business relationships between a customer and a service provider. You can define the above relationships using service agreement relationship functionality. This is a special type of SA relationship in which the head office is the service provider.

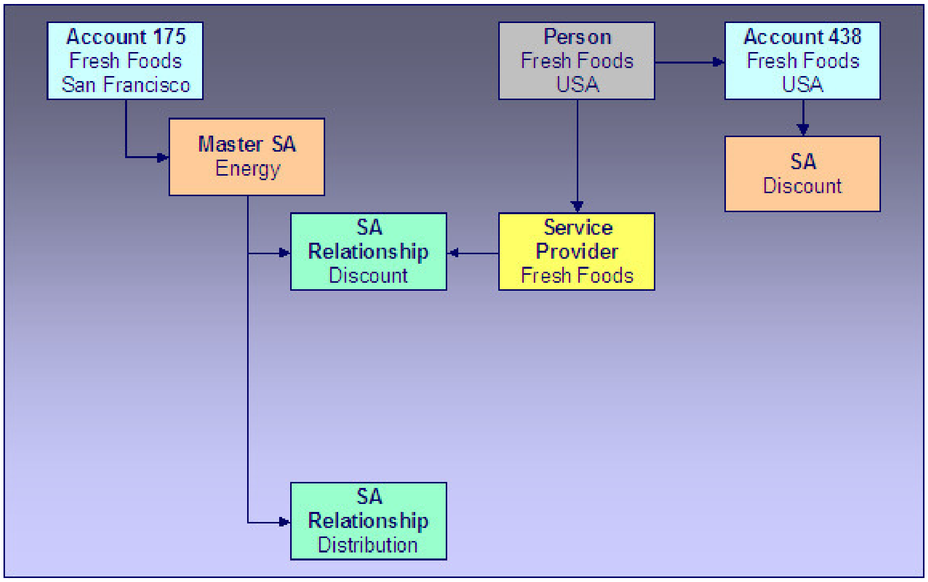
Examples Of Special Discounts

The following topics illustrate examples of how your organization may configure the system to handle this business functionality.

Example Using Aggregated Consumption

A customer, Fresh Foods has 10 stores each with its own account and electric service agreement, i.e., each store is billed separately. Fresh Food's head office decides to negotiate a group discount that applies to one or more of the individual stores.

In the following diagram, one store's account is shown.



Example Using Aggregated Consumption

Note the following:

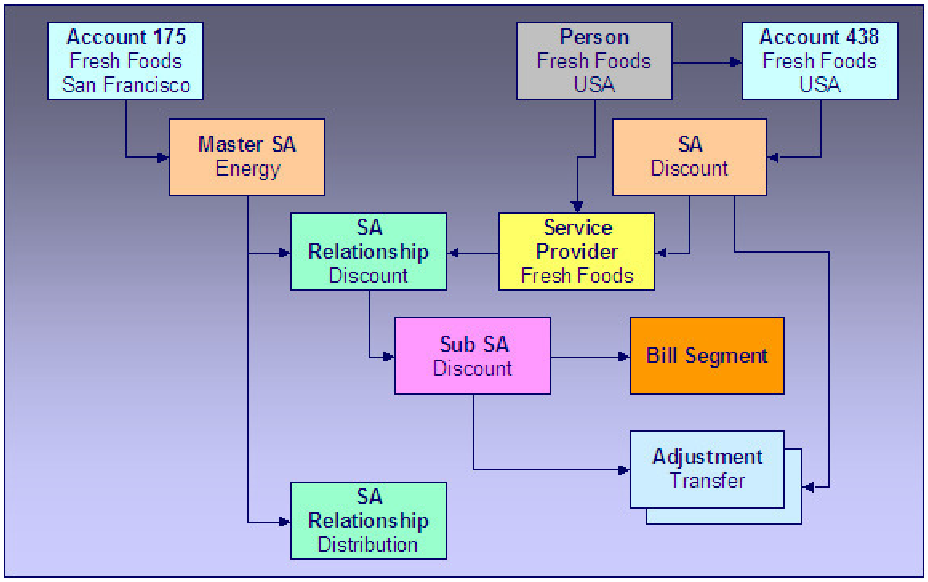
- An SA relationship is created for the energy SA covered by the negotiated terms.
- The service provider in the relationship is the head office.
- This relationship does not have a sub-SA because no additional billing services are provided for each individual covered service agreement.
- Discounts are calculated using the head office discount service agreement and affect only this service agreement and therefore only the head office's account.
- The master SA may have other deregulated relationships, such as the distribution relationship shown.

NOTE:

Aggregated consumption algorithm. In this scenario, the service provider's service agreement amalgamates the group's consumption and applies a rate to calculate the discount. This algorithm is not provided - you get to write this algorithm to meet your needs.

Example Using Site-by-site Discount

In this scenario, the discount is calculated on a site-by-site basis.



Example Using Site By Site Discount

Note the following differences between this example and the one where consumption is aggregated:

- Each service agreement participating in the discount has an SA relationship with a sub SA.
- Discounts are calculated on a site-by-site basis by the sub SA for each service agreement covered by the discount agreement.
- The discounts may be transferred to the head office service agreement using transfer adjustments. In this case, you may choose to not show the discount on the individual site's bills.

Setting Up The System To Enable Negotiated Terms

The above topics provided background information about how special negotiated terms could be supported in the system. The following discussion summarizes the various setup tasks alluded to above. These notes highlight the setup required in addition to that detailed in Setting Up SA Relationships Information.

Algorithm

If you use sub SAs to calculate charges / discounts for each covered service agreement individually and you want to transfer the charges / discounts to the service provider's service agreement at bill completion, you will need to set up a bill completion algorithm to do this.

Refer to the algorithm type BCMP-TR for more information.

SA Relationship Type

You will need an SA relationship type that will be used for each type of negotiated term or discount.

Service Provider

Set up a service provider using the principal party of the negotiated discount. For example, if a head office negotiates a special agreement for its sites, you should set up a new service provider representing the head office.

- **Billing Relationship** will depend on the type of negotiated terms you choose to set up. For terms where debits / credits are calculated for each covered service agreement individually, select We Bill For Them, Rate Ready. We are effectively billing on behalf of the principal party. For terms where the discount is calculated on an amalgamated SA there is no billing relationship because no additional billing services are provided for each covered service agreement.
- Select a **Payment Relationship** of Pay SPR Not Applicable if debits / credits to the sub-SA should not be transferred to the service provider. If you want to transfer charges at bill completion time using a bill completion algorithm, you should also select Pay SPR Not Applicable because you will not be creating adjustments to pay the principal party service at payment time or at billing time.
- Enter a **Transfer A/R Algorithm** to transfer the debit / credit from the sub-SA to the principal party's service agreement. If you do not want to transfer the debit / credit to the principally party's service agreement, you do not need to specify a transfer algorithm.

SA Type SA Relationship Type

Associate the SA relationship type setup for negotiated terms with the SA type(s) of the service agreements that are covered by the terms. The following are suggested values for the SA type SA relationship type:

- **Required**. Relationships of this type should not be required for the SA type because the negotiated terms will only cover selected service agreements with that SA type.
- **Gaps in SA Relationship** are Allowed if service agreements of this type do not have to have an SA relationship of this type with a service provider throughout the effective period of the SA.
- Add the service provider set up above to the list of valid service providers for this SA type and SA relationship type combination.

SA Type

Set up an SA type for the sub SAs that will be used to calculate discounts. This SA type should use the BCMP-TR bill completion algorithm if you want to transfer the discounts to the head office.

➤ **NOTE:**
TBFU Deregulated Relationship with WBFT Negotiated Terms Relationship. If you have the unusual situation where a master service agreement has a relationship with a TBFU service provider as well as a negotiated term "relationship" with a WBFT service provider, and you transfer charges / discounts from the discount agreement sub SA to the WBFT service provider, you will not be able to use the algorithms provided to transfer receivables to the service providers. This is because bill completion algorithms associated with the SA type of a master service agreement are executed before any bill completion algorithms associated with the SA types of sub service agreements. Consequently the TBFU XFER algorithm associated with the master service agreement will transfer all receivables from related sub SAs to the TBFU service provider before the BCMP-TR algorithm can transfer the discount agreement receivables. You will need to

modify the TBFU XFER bill completion algorithm that transfers receivables from the customer to the TBFU service provider to exclude the discount agreement receivables.

Defining Service Credit Options

Some companies allow their customers to participate in a special rewards program. The term "service credits" is used to describe a program that rewards customers for their business. The topics in this section provide details to help you set up the control tables required to support any service credit program that your company supports.

➤ **NOTE:**
Penalty Points. The service credits functionality is described in the documentation with the assumption that it is used for accumulating points to reward your customers. However, if your company has a business need to record penalty points for a customer, the service credits functionality may be used for that purpose as well.

The Big Picture Of Service Credits

The topics in this section provide background information about service credit functionality.

Service Credit Membership

Let's look at some examples of special programs that may use service credits functionality:

- Capital credits - When customers receive their utility service from a cooperative, they are considered "members" of the cooperative and may over time receive capital credit allocations from the cooperative based on their service history and the cooperative's profits allocated during that time period.
- Frequent flier miles - Perhaps your company has made an agreement with one or more airlines to allow customers to accumulate frequent flier miles for every x amount spent on service.
- Free pay-per-view movies - Maybe your cable service offers free pay-per-view movies under certain conditions. Using service credits, you can set up your system to accumulate the free pay-per-view movies and use the free movies to offset actual movies viewed by the customer.
- Any other type of loyalty program where the customer earns credits that may later be redeemed in some way.

To participate in a program such as those described above, the customer is linked to a service credit membership. The membership record provides the following functionality:

- It defines the accounts that are linked to the membership.

● **FASTPATH:**
Refer to Who are the Members? for information about linking persons and accounts to a membership.

- It defines a membership type, which controls certain behavior about the membership.

- It may define an external ID if the membership is associated with an external program, such as a frequent flier mile program.
- It may define a service agreement to use for miscellaneous financial transactions that may get created.
- It may define characteristics used to capture miscellaneous information about the membership.
- Over time, service credit events are created for a membership. The events indicate an amount that either adds or subtracts credit units (i.e., points, miles, movies, dollars, etc) for the membership.

- **FASTPATH:**
  Refer to The Big Picture of Service Credit Membership for more information about functionality related to a membership.

How Are Service Credits Earned?

Service credits may be monetary rewards for service or they may be non-monetary rewards such as free movies or frequent flier miles. In any case, how the membership earns the points or rewards depends on the business rules for the program you are offering.

A typical scenario is that the service credits are earned for a membership as a result of other services linked to the membership's accounts. For example:

- Perhaps free pay-per-view movies are earned when signing up for cable service. In this case, the pay-per-view movie membership is related to the membership account's cable service agreement. Refer to Service Credits Earned When Starting Service for more information.
- Perhaps one frequent flier mile is earned for every $10 spent on electricity. In this case, the frequent flier membership is related to the membership account's electricity service agreements (for example, electricity distribution, electricity retail and lamp service). Refer to service credits earned through billing for more information.
- For a capital credits membership, capital credit allocations are calculated based on the amount spent by the customer for standard service, for example electricity and/or gas service. In this case, the capital credit membership is related to the membership account's electricity and/or gas service agreements. For capital credits, a background process is used to calculate the allocated amounts. Refer to Allocating Capital Credit for more information.

It is also possible to earn service credits irrespective of other service for the membership's accounts. (Again, it depends on the business rules for the program you are offering.) For example, perhaps you offer 500 frequent flier miles for signing up for service with your company, regardless of the type of service chosen. In addition, assume that no additional miles are earned for ongoing service. In this example, there is no need to link the membership to any service agreements.

- **FASTPATH:**
  Refer to Service Agreements Contribute to a Membership for more information.

Each earned service credit amount is linked to the membership via a service credit event.

How Are Service Credits Redeemed?

Once service credits have been earned for a membership, how may a customer redeem these credits? The method by which the credits are redeemed depends on the business rules for the program you are offering. Here are some examples of how a service credit may be redeemed:

- For free pay-per-view movies, perhaps your customer's monthly cable bill is credited for any pay-per-view movies until all free movies are used. Refer to Service Credits Redeemed Through Billing for more information.
- For frequent flier miles, the information about the earned miles is exported to the appropriate airline. The miles are actually redeemed by the customer through the airline, not through your company. Refer to Interface Membership Information to a Third Party for more information.
- For a capital credit membership, the company periodically (maybe once a year) decides if credits should be redeemed (referred to as "retired") and if so, how much. The company runs a background process to calculate the "retirement" amount. When an amount is retired, the membership balance is reduced by the retirement amount and the amount is transferred to service agreements related to the membership. Refer to Capital Credit Retirement for more information.

> **NOTE:**
> **Tracking Membership Balances**. If service credits are redeemed via the system, your membership should probably be configured to keep track of a balance. Refer to Event Amounts May Contribute to a Balance for more information.

Each service credit amount redeemed through the system is linked to the membership via a service credit event.

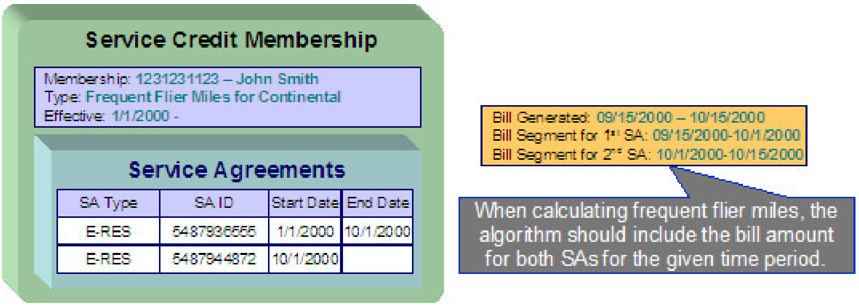
Service Agreements For a Membership

There are three types of service agreements that may be associated with a service credit membership. The following sections describe these types.

SAs Contribute to the Membership

As described in how are service credits earned, many memberships are related to specific service agreements for the membership's accounts. We refer to these service agreements as the SAs that contribute to the membership because often the service credit amounts earned for the membership are based on amounts spent by the customer for these services.

During the lifetime of a membership, service agreements that contribute to a membership may be stopped and other service agreements started. For example, perhaps you have a frequent flier membership that is related to electric service. Imagine that the customer starts out with a certain rate for electric service, but later decides to opt for a different type of rate that requires expiring the old service agreement and creating a new one. How does this affect your frequent flier mile calculation? It depends on how you design the algorithm that creates the frequent flier events. Essentially, the algorithm must cater for this situation. The following diagram illustrates the scenario.


Service Credit Membership
Membership: 1231231123 – John Smith
Type: Frequent Flier Miles for Continental
Effective: 1/1/2000 -
Service Agreements

SA Type	SA ID	Start Date	End Date
E-RES	5487936555	1/1/2000	10/1/2000
E-RES	5487944872	10/1/2000	

Bill Generated: 09/15/2000 – 10/15/2000
Bill Segment for 1st SA: 09/15/2000-10/1/2000
Bill Segment for 2nd SA: 10/1/2000-10/15/2000

When calculating frequent flier miles, the algorithm should include the bill amount for both SAs for the given time period.

The service agreements that contribute to a membership are not linked directly to the membership record. This would cause a maintenance burden, requiring links to be updated when service is stopped or started for applicable service agreements.

Rather, this link is indirect. The list of service agreement types is defined for the membership type. The system can determine which service agreements are "linked" to the membership by looking at the SA types for the membership type.

●

FASTPATH:
Refer to Determine The Types of Service Agreements That Contribute to the Membership for more information on designing your membership type to include appropriate SA types.

SA Used for Miscellaneous Transactions

For some memberships, you may need to define a special service agreement to use for miscellaneous transactions. For example:

If events created for a membership cause an adjustment to be created to affect the general ledger, rather than allowing the system to arbitrarily pick a service agreement to use for this adjustment, the SA to use should be indicated on the membership. Refer to An Event May Cause Other Actions to Occur for more information.

Membership Fee SAs

For some memberships, a membership fee may be applicable. A special service agreement is used to hold the fee. This service agreement is not linked directly to the membership, but is simply a service agreement linked to one of the membership's accounts.

➤

NOTE:
Some fees may be refundable. The refunding of a fee must be handled by an algorithm. Refer to SAST-RF for information about the algorithm type provided with the base product.

Designing Your Service Credit Options

This section helps you to determine how to design your service credit membership types and service credit event types.

Designing Your Membership Types

This section discusses the options to consider when designing your service credit membership types.

First consider the type of unit that your membership's service credit events represent.

- Is it related to a currency? If so, ensure that your currency is correctly defined on the currency page.
- Is it a non-currency unit, such as movies, points or miles? If so, you need to define the unit on the credit unit page.

Next, consider other behavior that your membership may exhibit.

- Should your membership calculate an overall balance ?
- Will miscellaneous financial transactions be created for you membership over its lifetime? If so you may need to link a service agreement to your membership.

- Should events linked to your membership reference a fiscal year ?

The table below illustrates three types of memberships: one for capital credits, one for frequent flier miles and one for free pay-per-view movies.

Membership Type	Description	Unit Type / Currency or Credit Unit	Has Balance?	Require SA?	Fiscal Year?
STDCAPCR	Standard capital credit membership	Currency / $	Yes	Yes	Yes
FFDELTA	Delta frequent flier miles	Credit Unit / Miles	No	No	No
FREEPPV	Free pay-per-view movies	Credit Unit / Movies	Yes	No	No

The following points explain the settings for each membership above:

- The capital credit membership uses a currency of US dollars for its units. Over time, capital credits are allocated and retired. The overall balance of the credits and debits should be calculated and displayed for information purposes. It is common for the allocation and retirement of capital credits to affect the GL, and as a result, a membership SA is required for posting these financial effects. Finally, in a capital credit situation, the allocation is typically related to a specific fiscal year. When calculating and displaying balances, the balance for each fiscal year must also be available. As a result, the fiscal year must be set to required.
- For the frequent flier membership type, a separate membership type must be created for each different airline. This is because a separate membership must exist for each separate airline in order to keep track of the accumulated miles correctly. The membership type does not have a balance because the accumulated miles are interfaced to the airline and the airline keeps track of the balance. In this example, no SA is required because an assumption is being made that the creation of frequent flier miles does not affect the GL. (Accumulating miles is no liability or expense for the company. The miles are simply accumulated on behalf of a third party.) Finally, the frequent flier miles do not need to indicate a fiscal year.
- For the free pay-per-view movies membership, we assume that the credits are redeemed from within the system. For example, perhaps a calculation rule calculation algorithm or pre-processing calculation rule redeems the free pay-per-view movies over time as customers are billed for the movies. As a result, the membership should have a balance. In this example, no SA is required because an assumption is being made that any affect on the GL is posted at the time the free movies are redeemed (i.e., when calculating a bill). (This is just an example. It's possible that you may want to post to the GL when free movies are accumulated to mark a payable for the company.) Finally, free pay-per-view movies do not need to indicate a fiscal year.

More options must be considered for each membership type.

Consider Whether the Membership Should Indicate Subcategories

For some types of memberships, the amount of each service credit event is further grouped by a subcategory. Refer to Events May Indicate a Subcategory for more information.

Use subcategories if multiple subtypes of credits may be accumulated and redeemed and if balances need to be tracked for each subcategory.

For our sample membership types, only the capital credits type should indicate subcategories.

Membership Type	Description	Subcategories
STDCAPCR	Standard capital credit membership	Distribution Transportation
FFDELTA	Delta frequent flier miles	

FREEPPV	Free pay-per-view movies

Determine the Types of Service Agreements Linked to the Membership

Memberships typically exist to reward customers for participating in standard service with the company. Refer to How Are Service Credits Earned? for more information. If memberships of this type are related to standard service for your company, determine the SA types for these service agreements.

NOTE:
Standard Service SA Types. These service agreement types are probably different from the type of service agreement you may link as the membership SA. The membership SA is used for miscellaneous charges like general ledger posting. These SA types are related to electricity service, gas service, etc.

For our three examples, assume that the company provides electric service and cable. The capital credits are related only to electric service, frequent flier miles are accumulated for a combination of services, and free pay-per-view movies are related only to cable service.

Membership Type	Description	SA Types
STDCAPCR	Standard capital credit membership	E-RES (Electric Residential) AL-RES (Area Lighting)
FFDELTA	Delta frequent flier miles	E-RES (Electric Residential) AL-RES (Area Lighting) CABLE -RES (Cable Residential)
FREEPPV	Free pay-per-view movies	CABLE-RES (Cable Residential)

Consider Special Functionality Needed When Adding, Activating or Inactivating a Membership

Should a letter or bill message be generated when a membership is created or activated? Should anything happen when a membership is inactivated? If so, you may need to define one or more algorithms to be plugged in on the membership type.

For our examples, let's say that a bill message is generated when a frequent flier or free pay-per-view membership is created. Let's say that a letter is generated when a capital credits membership is activated. We also assume that when a capital credits membership becomes inactive, any outstanding balance is redeemed. For the frequent flier miles and pay-per-view memberships, we assume nothing special occurs when the membership becomes inactive.

FASTPATH:
Refer to Lifecycle of a Membership for more information about the various status values for a membership.

Membership Type	Description	Algorithm System Event	Algorithm

STDCAPCR	Standard capital credit membership	Membership Activation	Send Letter
		Membership Inactivation	Redeem Balances
FFDELTA	Delta frequent flier miles	Membership Creation	Generate Bill Message
FREEPPV	Free pay-per-view movies	Membership Creation	Generate Bill Message

NOTE:

Sample Algorithms. The base package does not provide sample algorithms for all the above examples. Refer to Service Credit Membership Type - Algorithm for more information about the algorithms provided with the system.

Determine Whether The Membership Requires Additional Information

Is there any information about your membership that you need to capture that is not already provided by the base system logic? If the answer is yes, you may need to define characteristics for your membership. Use the characteristic collection on the membership type to define the types of characteristics allowed for memberships of this type. You may also define default values for you membership's characteristics.

Designing Your Service Credit Event Types

Now that you have designed your membership types, you need to design the types of service credit events that may be created for your membership.

In many cases, a credit event should cause additional functionality to occur. Algorithms are executed when an event is completed and when an event is canceled and are used to perform additional functionality. The following points illustrate possible algorithms that may be needed when a credit event is completed.

- Validate the event as compared to other events. For example, perhaps a new event should never cause the overall membership balance to fall below zero. You could use an algorithm on the service credit event type to check this condition.
- Create an adjustment that posts to the general ledger. Your credit event may not affect the customer's balance when created, but perhaps it should have an effect on the general ledger. For these types of credit events, you may need to create an adjustment to post to the GL.
- Create adjustments to affect the customer's balance. When "redeeming" a credit, you may need to transfer a monetary amount to one or more of the customer's service agreements.
- Create a bill message. Perhaps when a credit is accumulated, you want to inform the customer via a message on the bill indicating the credit amount. A temporary bill message is added to one of the membership's accounts.

NOTE:

One Account Message of the Same Message Type. The temporary bill message collection on the account allows only one bill message of the same bill message type. If it's possible for multiple types of events to be generated for the same account, consider creating a different bill message type for each event type.

- Stamp a batch code and batch run number onto the event. This would be used when your event information needs to be interfaced to an external system.

The following points illustrate possible algorithms that may be needed when a credit event is canceled.

- Validate the event as compared to other events. For example, perhaps canceling an event should never cause the overall membership balance to fall below zero.
- Cancel adjustments that may have been created when the event was completed.

To illustrate examples of when to use some of the algorithms above, we'll design event types for the membership types designed above.

Designing Capital Credit Event Types

The following event types illustrate typical events for a capital credits membership.

SC Event Type	Description	Membership Type	Algorithm System Event	Algorithm
ALLOCATECCR	Capital credit allocation	STDCAPCR	Event Creation	Create Simple Adjustment
			Event Creation	Generate Bill Message
			Event Cancellation	Validate Balance Not < Zero
			Event Cancellation	Cancel Related Adjustments
RETIRECCR	Retire capital credit (apply to customer's balance)	STDCAPCR	Event Creation	Validate Balance Not < Zero
			Event Creation	Create Adjustments to Affect Customer's Balance
			Event Cancellation	Cancel Related Adjustments
FORFEITCCR	Forfeit capital credit (retirement not applied to customer's balance)	STDCAPCR	Event Creation	Validate Balance Not < Zero
			Event Creation	Create Simple Adjustment (affect GL only)
			Event Cancellation	Validate Balance Not < Zero
			Event Cancellation	Cancel Related Adjustments

These event types assume the following:

- When credit events are allocated, the customer is notified via a bill message, the general ledger is affected so a simple GL only adjustment is created. If an event of this type is canceled, any adjustments that were created should be canceled. The event amount should always be positive, so checking that the membership balance does not fall below zero is only checked for event cancellation.
- When capital credits are retired, it's possible that the full membership balance is not applied to the customer's balance. For the portion of the retirement that does affect the customer's balance, you need an algorithm that applies the credits to the customer's service agreements via adjustments. Cancellation of this event should cause any related adjustments to

be canceled. It's assumed that the amount of this event is a credit so checking that the membership balance does not fall below zero is only checked for event completion.

- For the portion of the retirement that is not applied to the customer's balance, the event amount should simply affect the GL so a GL only adjustment is created. Cancellation of this event should cause any related adjustments to be canceled. It's assumed that the amount of this event is a credit so checking that the membership balance does not fall below zero is only checked for event completion.

> **NOTE:**
> **Service credit event types are independent of subcategories.** A capital credits membership typically uses subcategories. When events are created for different subcategories, the same service credit event type may be used. As a result, all subcategories use the same event completion and event cancellation algorithms.

Designing Frequent Flier Event Types

The following event types illustrate typical events for a frequent flier membership:

SC Event Type	Description	Membership Type	Algorithm System Event	Algorithm
ADDMILES	Add miles to the membership	FFDELTA	Event Creation	Generate Bill Message
			Event Creation	Populate Batch Information
			Event Cancellation	Event may not be canceled.

This event type assume the following:

- A new event should generate a bill message.
- Information about the event amount should be interfaced to an external system so batch information should be populated when the event is completed.
- Events of this type may not be canceled because the information is interfaced to an external system. Rather, to reverse an event, simply create a new event whose amount is a credit. This credit amount is also interfaced to an external system.

> **NOTE:**
> **Sample Algorithm.** The system does not provide a sample cancellation algorithm that prevents the event from being canceled.

- There is no validation to ensure that the balance does not fall below zero. Recall that this membership was defined as not requiring a balance.

Notice that only one type of event has been defined for this membership. That is because the credits for this membership are not redeemed via this system. Rather they are accumulated on behalf of an external system.

Designing Pay-per-view Event Types

The following event types illustrate typical events for a free pay-per-view movies membership:

SC Event Type	Description	Membership Type	Algorithm System Event	Algorithm
ADDPPV	Free pay-per-view movies	FREEPPV	Event Creation	Generate Bill Message
			Event Cancellation	Validate Balance Not < Zero
REDEEMPPV	Redeem pay-per-view movies	FREEPPV	Event Creation	Validate Balance Not < Zero
			Event Creation	Generate Bill Message

These event types assume the following:

- Any type of new event should generate a bill message.
- For adding free movies, it is assumed that the quantity is positive so when the event is cancelled, the algorithm verifies that the balance does not become negative.
- For redeeming free movies, it is assumed that the quantity is negative so when the event is completed, the algorithm verifies that the balance does not become negative.
- These events do not affect the general ledger and they do not directly affect the customer's balance so no adjustment algorithm is needed.

How Are Service Credit Events Created?

Now that you have designed the behavior of your service credit events, an important issue is to determine how these events are created. Consider your business practice for each type of membership.

Let's use our sample memberships to work through different ways that you may create service credits.

Service Credits for Capital Credit Memberships

For a capital credit membership, capital credit allocations are calculated once a year based on billing history and the cost of service. To accomplish this, a background process calculates the amount and creates the service credits. Refer to allocating capital credits for a sample process provided with the base package.

Credits are redeemed via the retirement process. The company determines when to retire capital credits based on analysis of their financial situation. This retirement process is also handled by a background process. Refer to capital credit retirement for a process provided with the base package.

For capital credits memberships, special functionality is required when a member dies. The capital credits are considered part of the person's estate and may need to be retired and applied to a beneficiary's account. This process depends on the company's business practice. However, typically, the membership status would change to inactive so that new capital credit allocations are not created for the membership. The system provides a sample inactivation algorithm called SCMI-RB that transfers part or all of the outstanding credit balance to the member's service agreements. From there, a user can cut a check to the beneficiary.

NOTE:
Some Credits Are Never Retired. For many cooperatives, some types of allocated credits are never retired. Refer to Partial Retirement for more information.

Service Credits for Frequent Flier Memberships

In our example, frequent flier miles were related to both the electricity and phone service. Let's assume that frequent flier miles are accumulated every $x spent on electricity and phone service. In this example, a bill completion algorithm creates service credits based on the bill segment amounts for these service agreements. Refer to Service Credits Earned Through Billing for more information.

As mentioned before, the frequent flier miles are not redeemed using this system, but are interfaced to a third party for redemption.

Service Credits for Free Movies Memberships

NOTE:
Sample Algorithms. No base package algorithm types are provided to support the logic described in this example.

In this scenario, let's suppose that a customer receives three free pay-per-view movies when signing up for new service. To handle this, perhaps an SA creation algorithm creates the membership and the service credits when the service is started. Or perhaps you want to wait until the first bill is generated and a bill completion algorithm is used to generate the first credit. It depends on your business practice.

For redeeming the free pay-per-view movies, it is assumed that the movies are credited during billing after it is determined that the customer has been billed for a movie. The number of movies used for the membership is reduced until all the free movies are used.

FASTPATH:
Refer to Service Credits Redeemed Through Billing for more information.

Setting Up Service Credit Options

Setting Up Credit Units

Credit unit is used for service credit membership types whose events record non-monetary units. Open **Admin** > **Service Credit** > **Credit Unit** to set up credit units.

Description of Page

The following fields display for each credit unit:

Credit Unit The unique identifier of the credit unit.

Description The description of the unit. This also acts as a label for the unit when displaying information about a service credit event.

Symbol / Label Position Indicates whether or not the label for this credit unit appears as a Prefix or as a Suffix to the service credit event amount.

Decimal Positions Indicates the number of decimal positions used for this credit unit. This information should be used by any algorithm or background process that creates a service credit event to determine how to store the event amount. It is also use to correctly display the service credit amounts.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference CI_CR_UNIT.

Setting Up SC Membership Inactive Reasons

The service credit membership inactive reason must be specified when the status of a service credit membership changes to inactive. Open **Admin** > **Service Credit** > **Service Credit Membership Inactive Reason** to set up service credit inactive reasons.

Description of Page

The following fields display for each inactive reason:

Inactive Reason The unique identifier of the service credit membership inactive reason.

Description The description of the inactive reason.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference CI_SCM_NCTV_RSN.

Setting Up Service Credit Membership Types

SC Membership Type - Main

Service credit memberships have a membership type. Open **Admin** > **Service Credit** > **Service Credit Membership Type** > **Add** to define the membership type.

Description of Page

Enter a unique **Service Credit Membership Type** and **Description** for each membership type.

Use the **SA Requirement** flag to indicate whether a miscellaneous SA must be linked to memberships of this type. The possible values are SA Required and SA Not Allowed .

> **NOTE:**
> The value defaults to SA Required.

Use the **Fiscal Year Requirement** flag to indicate whether events linked to memberships of this type should indicate a fiscal year. The possible values are Fiscal Year Required and Fiscal Year Not Allowed.

➤ **NOTE:**
Default Note. The value defaults to Fiscal Year Not Allowed.

Use the **SCM Event Balance** flag to indicate whether or not a balance of service credit events linked to memberships of this type should be calculated and displayed. The possible values are Has a Balance and No Balance.

➤ **NOTE:**
Default Note. The value defaults to Has a Balance.

Use the **SC Membership Type Unit** flag to indicate whether or not the unit for the event amounts for events linked to memberships of this type is Currency or Credit Unit. When the unit is currency, indicate the **Currency Code**. When the unit is credit unit, indicate the **Credit Unit**.

If events linked to memberships of this type must indicate a subcategory, enter a valid **Subcategory Name** and **Description** for each subcategory applicable to this membership.

● **FASTPATH:**
Refer to Events May Indicate a Subcategory for more information.

Use the **SA Types** grid to indicate the types of service agreements that contribute to memberships of this type. Indicate the **CIS Division** and **SA Type** for each type of service agreement that is related to a membership of this type.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference CI_SCM_TYPE.

SC Membership Type - Algorithm

Open **Admin** > **Service Credit** > **Service Credit Membership Type** > **Search** and navigate to the **Algorithm** page to define any algorithms that are associated with a membership type.

Description of Page

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 (descriptions of all possible events are provided below).
- 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.

The following table describes each **System Event** for which you can define algorithms.

System Event	Optional / Required	Description
Membership Creation	Optional	Use this system event for algorithms that are executed when a membership is created.

		Click here to see the algorithm types available for this system event.
Membership Activation	Optional	Use this system event for algorithms that are executed when a membership becomes active. Click here to see the algorithm types available for this system event. ➤ **NOTE:** **Creating Memberships In Active Status**. A membership may be created in either pending status or in active status, based on your business practice. For memberships created in active status, the system executes the membership creation algorithms and the membership activation algorithms.
Membership Inactivation	Optional	Use a system event of Membership Inactivation for algorithms that are executed when a membership becomes inactive . Click here to see the algorithm types available for this system event.

SC Membership Type - Characteristics

To define characteristics that can be defined for service credit memberships of a given type, open **Admin** > **Service Credit** > **Service Credit Membership Type** > **Search** and navigate to the **Characteristics** page.

Description of Page

Use the characteristics collection to define characteristics that can be defined for service credit memberships of a given type. Use **Sequence** to control the order in which characteristics are defaulted. Turn on the **Required** switch if the **Characteristic Type** must be defined on service credit memberships of a given type. Turn on the **Default** switch to default the **Characteristic Type** when service credit memberships of the given type are created. Enter a **Characteristic Value** to use as the default for a given **Characteristic Type** when the **Default** switch is turned on.

Setting Up Service Credit Event Types

Service credit events created for a service credit membership have an event type. Open **Admin** > **Service Credit Event Type** > **Add** to set up service credit event types.

Description of Page

Enter a unique **Service Credit Event Type** and **Description** for each event type.

Indicate the **Service Credit Membership Type** to which this event type belongs.

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 (descriptions of all possible events are provided below).
- 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.

The following table describes each **System Event** for which you can define algorithms.

System Event	Optional / Required	Description
Event Completion	Optional	Use a system event of Event Completion for algorithms that are executed when an event is completed. Refer to An Event May Cause Other Actions to Occur for more information. Click here to see the algorithm types available for this system event.
Event Cancellation	Optional	Use this system event for algorithms that are executed when an event is canceled. Click here to see the algorithm types available for this system event.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference CI_SC_EVT_TYPE.

Service Credit Examples

In this section, we provide examples of how to define your control tables to support functionality related to different types of service credit memberships. While your company may not define your environment exactly the same way, this section should help solidify your understanding of how to set up your company's service credit options.

Defining Control Tables for a Refundable Fee

If your membership requires payment of a fee that is refundable, you must define an SA type to use for the refundable fee.

Adjustment Type for Refundable Fee

This example assumes that the fee is set up similarly to a deposit. The adjustment used to charge the fee would affect the current balance, but not the payoff balance or the general ledger because the fee is not considered a "receivable", rather it is an amount collected and held for the customer.

Create an adjustment type for levying the fee. Indicate the fee amount and indicate the adjustment FT creation algorithm ADJT-CA, which affects the current balance, but not the payoff balance or the general ledger.

➤ **NOTE:**
Adjustment Type Profiles. Be sure to add this adjustment type to an appropriate adjustment type profile and ensure this profile is linked to your SA type.

SA Type for Refundable Fee

Create an SA type to use for the fee. This SA type should be marked as not billable.

Indicate the appropriate adjustment type profile that includes the adjustment type to levy the fee.

Indicate an appropriate payment segment type that references the PSEG-NM payment segment FT algorithm. This algorithm affects the payoff amount and current amount by the payment amount, which should cause the current amount to become zero and the payoff amount to become a credit for the fee amount when the fee is paid.

Start Option for Refundable Fee

For this SA type, define a start option that causes an adjustment to be levied as part of the start service process.

● **FASTPATH:**
Refer to Including The Membership Fee on campaigns and packages for more information about levying the fee via the order transaction.

Algorithm for Refunding the Fee

The base product provides an algorithm type SAST-RF that refunds a fee when service agreements that contribute to the membership are stopped. You must create an algorithm for this algorithm type and enter the SA type created above as an input parameter.

This SA stop algorithm must be plugged in on all SA types that you defined for the membership type.

Defining Control Tables for a Nonrefundable Fee

If your membership requires payment of a fee that is not refundable, you can set this up in two ways:

- You can create a special SA type to handle charging the fee. For this SA type, define a start option that causes an adjustment to be levied as part of the start service process. This adjustment contains your fee. You may use this option if the membership is related to multiple types of services and not all services need to be present in order to create the membership.
- You can levy an adjustment on one of the other service agreements that is being started. To do this, you would use a start option to define an adjustment to be levied as part of the start service process. You may use this option if the membership is related to a single service (for example, free pay-per-view movies).

For our example, we will set up the SA for the nonrefundable fee with a separate 'fee' SA type.

Adjustment Type for Nonrefundable Fee

Create an adjustment type for levying the fee. Indicate the fee amount and indicate the adjustment FT creation algorithm ADJT-NM, which affects the current balance and payoff balance by the adjustment amount, and affects the general ledger.

➤ **NOTE:**
Adjustment Type Profiles. Be sure to add this adjustment type to an appropriate adjustment type profile and ensure this profile is linked to your SA type.

SA Type for Nonrefundable Fee

Create an SA type to use for the fee. This SA type should be marked as not billable.

Indicate the appropriate adjustment type profile that includes the adjustment type to levy the fee.

Indicate an appropriate payment segment type that references the PSEG-NM payment segment FT algorithm. This algorithm affects the payoff amount and current amount by the payment amount, which causes the current amount to become zero and the payoff amount to become a credit for the fee amount when the fee is paid.

Start Option for Nonrefundable Fee

For this SA type, define a start option that causes an adjustment to be levied as part of the start service process.

● **FASTPATH:**
Refer to Including The Membership Fee on campaigns and packages for more information about levying the fee via the order transaction.

Defining an SA Type for Miscellaneous Transactions

Does your membership require a service agreement to support miscellaneous transactions? If so, you need to consider the SA type to use for this service agreement. This SA type should be marked as not billable.

If you choose to use the order transaction to set up the membership, this SA type must be defined on the algorithm that creates the membership.

It is possible that you may require a start option for this SA type, for example if you want to define a characteristic for service agreements of this type.

Using Campaigns/Packages to Set Up Membership

If enrollment in a membership is a common occurrence for your customers when starting service, you should consider using the order page to start service for the customer and to create the membership as well.

NOTE:
This section only makes sense if you are familiar with the Sales and Marketing chapter.

The recommendation is to use a question/miscellaneous field to ask the customer service representative to indicate the appropriate membership type. Algorithms validate this membership type and use it to create a membership of that type.

This section walks you through how to set up the campaign / package required to support this.

Column Reference for Membership Type

In order to ask the customer service representative to define an appropriate service credit membership type, you must define a column reference for the membership type.

Add a new column reference with the following information:

- Column Reference: SCM-TYPE
- Description: SC Membership Type
- FK Reference: SCM TYPE
- Long Description: Service Credit membership type to use when creating a service credit membership at start time.

NOTE:
Column Reference Algorithms. This column reference indicates a validation algorithm and a posting algorithm. However, we have not defined them yet so for now simply save this information.

Algorithms to Create Membership via Order

The base product provides two algorithm types to support the creation of a membership record via the order page: a validation algorithm type and a posting algorithm type.

Validate Membership Information

This algorithm is a column reference validation algorithm that checks that an input membership type is valid. Refer to CRVL-ME for more information. You must define an appropriate algorithm for this algorithm type, and on that algorithm you must define the column reference used to identify the membership type. For example:

- Algorithm: VAL MEM TYPE
- Description: Validate SC Membership Type
- Algorithm Type: CRVL-ME
- Parameter1: (Column Reference for Membership Type): SCM-TYPE

Post Membership Information

This algorithm is a column reference posting algorithm that creates a membership using the membership type indicated by the user. Refer to CRPS-ME for more information. You must define an appropriate algorithm for this algorithm type, and on that algorithm you must define the column reference used to identify the membership type. For example:

- Algorithm: CREATE MEMBRSHP
- Description: Validate SC Membership Type
- Algorithm Type: CRPS-ME
- Parameter1: (Column Reference for Membership Type): SCM-TYPE
- Parameter2: (Service Credit Membership Status): 10 -Pending or 20 -Active. Refer to Lifecycle of a Membership for more information.
- Parameters3-5: (Division, SA Type, Start Option): Indicate the information needed to create an SA for miscellaneous transactions.

> ➤ **NOTE:**
> This algorithm first looks for an existing service agreement of this division / SA type linked to the membership's accounts. If one exists, it uses that SA to link to the membership. If an SA does not exist, it creates one with the input division, SA and (optional) start option.
>
> **Update the Column Reference.** Now that you have defined the validation and posting algorithms, return to your column reference for membership type and define the algorithms there.

Although your company may support multiple types of memberships, this column reference and its algorithms have been designed such that only one column reference for membership type would be needed to set up any type of membership. Although the posting algorithm for the membership type column reference indicates an SA type to use for miscellaneous transactions, a service agreement is only created and linked if your membership type indicates that an SA is required. As a result, you may use the same column reference for both memberships that require an SA and those that don't require an SA. However, if you have different membership types that require an SA and each uses a different SA type or a different start option for the membership SA, you need to define more than one posting algorithm and, as a result, more than one column reference.

Define a Campaign for Creating a Membership When Starting Service

Many factors must be considered when designing your campaigns and packages. The possible creation of a membership when using the order page is simply another factor to consider.

If you plan to define a question/miscellaneous field to capture a service credit membership type, the available packages linked to the campaign should be ones that are related to memberships. For example, if your membership is related to electric service, it doesn't make sense to create a membership for a campaign designed to generate a one-time charge.

When defining a question/miscellaneous field, you must indicate its applicability. Consider whether a membership type is required, optional or only applicable for certain packages. This helps ensure that your users capture this information when appropriate.

Including the Membership Fee

You must determine the best way to setup your campaign/package to handle the levying of your membership fee, if applicable. Consider some of the following questions.

- If the membership has a fee, is it a refundable fee or a non-refundable fee?
- Is the fee always applicable for the membership? Or is it waived under certain conditions? For example, maybe the fee is applicable if a customer signs up for a single service, but the fee is waived if the customer signs up for two or more services.

The SCM type column reference and question/miscellaneous field are set up to ask the user what type of membership to create. Because the applicability of the fee may differ for each membership type, you should carefully consider the campaign / package setup to levy the fee correctly.

- If a fee is always applicable for a membership, you may consider creating a membership creation algorithm that creates a fee SA with a start option to levy the fee when the membership is created.

  ➤ **NOTE:**
  Sample Algorithm. The base product does not provide a membership creation algorithm to do this.

- If the fee is not always applicable for a membership, you must determine when the fee is applicable. The recommendation is to include the fee SA type in the SAs-to-create collection for the appropriate package. For example,
  - If the fee is applicable when the customer signs up for a single service, you should define a package for each single service that includes one SA to create for the single service and the fee SA as another SA to create.
  - If the fee is waived when the customer signs up for two or more services, you should define a package for each combination of the multiple services. These packages do not include the fee SA in the SAs to create.
- If the customer signing up for service is a former customer who has returned, perhaps the fee was paid earlier when the customer originally had service. In this case, maybe your business practice is to waive the fee at this time. To do this, you should set up a question/miscellaneous field for the user to mark that this is a returning customer. Based on the answer to this question, perhaps only packages that do not levy any fee are eligible for selection.

Defining Another Person for Your Account

It is common for a capital credit membership to define more than one person for the account being turned on and linked to a membership. A typical example is a married couple. Both spouses are indicated on the account and financially responsible persons and as a result, both are considered members.

Column reference algorithms have been provided by the base product to enable linking a second person to your account via the order page.

This section walks you through how to set up the column reference and campaign to support this.

➤ **NOTE:**
Linking A Second Person to the Account. This logic is not restricted to service credit functionality. Any campaign may be designed to include the ability to link a second person to the account being started.

Column References for Additional Person

In order to ask the customer service representative to link an additional person for the account, you must define several column references to use as questions/miscellaneous fields.

- We should assume that the person may already exist in the database. As a result, a question to record the person ID is needed.
- If the person does not already exist, the user should capture the person's name, an ID type and an ID number. Questions for these three fields are needed. The system uses this information to create a new person.

- Whether we are using an existing person or creating a new one, the person's link to the account must include an account relationship type. A question to record the account relationship type is needed.

Add a new column reference for person ID:

- Column Reference: PERSON-ID
- Description: Person ID
- FK Reference: PER
- Long Description: Person ID of the additional person to link to the order's account.

Add a new column reference for person name:

- Column Reference: PERSON-NAME
- Description: Person Name
- FK Reference: not applicable
- Long Description: Name of a new person to create.

Add a new column reference for identifier type:

- Column Reference: PER-ID-TYPE
- Description: Identifier Type
- FK Reference: ID TYPE
- Long Description: Identifier type for the primary ID of the additional person to link to the order's account.

Add a new column reference for identifier number:

- Column Reference: PER-ID-NUM
- Description: Person ID Number
- FK Reference: not applicable
- Long Description: Identifier number for the primary ID of the additional person to link to the order's account.

Add a new column reference for account relationship type:

- Column Reference: ACCT-REL-TYPE
- Description: Account Relationship Type
- FK Reference: ACCT REL
- Long Description: Relationship type to use for the link between the additional person and the order's account.

> ➤ **NOTE:**
> **Column Reference Algorithms**. One of the column references above must indicate a validation algorithm and a posting algorithm. However, we have not defined them yet so for now simply save this information. We recommend using the account relationship type record because that is used for all additional persons.

Algorithms to Link Additional Person via Order

The base product has provided two algorithm types to support the linking of an additional person to the account via the order page: a validation algorithm type and a posting algorithm type.

Validate Addition Person Information

This algorithm is a column reference validation algorithm which checks that either a person id or person name, a valid ID type and ID number are provided and that a valid account relationship type is provided. Refer to CRVL-PE for more information. You must define an appropriate algorithm for this algorithm type and on that algorithm you must define the column reference used for the five fields required for this validation. For example:

- Algorithm: VAL ADD PER
- Description: Validate Additional Person Info
- Algorithm Type: CRVL-PE
- Parameter1: (Column Reference for Person ID): PERSON-ID
- Parameter2: (Column Reference for Person Name): PERSON-NAME
- Parameter3: (Column Reference for Person ID Type): PER-ID-TYPE
- Parameter4: (Column Reference for Person ID Number): PER-ID-NUM
- Parameter5: (Column Reference for Account Relationship Type): ACCT-REL-TYPE

Post Addition Person

This algorithm is a column reference posting algorithm that may link an existing person to the order's account or create a new person and link that person to the order's account. Refer to CRPS-PE for more information. You must define an appropriate algorithm for this algorithm type, and on that algorithm you must define the column reference used for the five fields required for this logic. For example:

- Algorithm: LINK ADDNTL PER
- Description: Link additional person to order's account
- Algorithm Type: CRPS-PE
- Parameter1: (Column Reference for Person ID): PERSON-ID
- Parameter2: (Column Reference for Person Name): PERSON-NAME
- Parameter3: (Column Reference for Person ID Type): PER-ID-TYPE
- Parameter4: (Column Reference for Person ID Number): PER-ID-NUM
- Parameter5: (Column Reference for Account Relationship Type): ACCT-REL-TYPE

NOTE:
Update the Column Reference. Now that you have defined the validation and posting algorithms, return to your column reference for account relationship type and define the algorithms there.

Design a Campaign to Include Linking an Additional Person

Any campaign related to a specific account may include the questions and miscellaneous fields defined here to create/link an additional person to the account.

Simply create entries in the questions and miscellaneous fields collection to prompt the user to ask for the required information. A question should exist for each column reference created above. The following table illustrates a possible setup.

Seq	Description	Prompt on Order	Column Reference	Dependency
10	Account relationship type for additional person.	If you would like to link another person to this	ACCT-REL-TYPE	Must have account

		account, please enter an account relationship type.		
20	Person ID for additional person	If the additional person already exists in the database, enter the person id.	PERSON-ID	Must have account
30	Person name for additional person	If you would like to create a new person, please enter the person name.	PERSON-NAME	Must have account
40	ID type for additional person	Please enter the ID type of the new person.	PER-ID-TYPE	Must have account
50	ID Number for additional person	Please enter the ID number of the additional person.	PER-ID-NUM	Must have account

Service Credits Earned When Starting Service

For some memberships, you may want to add service credits when starting the program. For example, the customer gets three free pay-per-view movies when signing on for cable service. Or a customer receives 500 frequent flier miles for signing up for a combination of gas and electric service.

There are various ways that you can accomplish this. You should work with your implementers to consider the various options to determine the method that best suits your business practice.

- You could use an SA activation algorithm to create a membership and an event for the initial points. Use this method if the points are related to a single type of service agreement and the points are earned automatically when starting service (i.e., without any human decision to be made).
- You could use a membership creation or membership activation algorithm to generate a service credit event automatically. This method assumes that the decision to create the membership has been made and that the free initial points are always earned for this type of membership (i.e., regardless of the type of service created).
- You could use questions and miscellaneous fields for a campaign/package to determine a customer's eligibility for participation in the membership and for the initial free points. A column reference posting algorithm could create the membership and/or the service credit event for the free points based on the answers to the questions.
- Perhaps the initial free points are only earned after the first bill is generated. Use a bill completion algorithm to generate the initial points.

You may think of other plug-in spots that could be used to generate free initial points based on your business needs.

Service Credits Earned Through Billing

For some memberships, you may accumulate points as a result of billed amounts for other services. For example, perhaps your customers earn one frequent flier mile for every $10 spent on the combined electricity, gas and water bills each month.

To accomplish this, you must design an algorithm to be executed at billing time. There are various plug-in spots executed at billing time that you may use, but the recommended plug-in is a post completion algorithm on the customer class. This plug-in is executed after all bill segments are frozen and most of the completion logic has occurred.

Your algorithm should determine the service agreements that contribute to the membership and calculate the service credit amount for those service agreements bill segments.

This algorithm should also consider what to do when bill segments that contributed to the event are canceled. The algorithm provide with the base product CBCM-SC checks to see if any canceled bill segments are referenced on any previous events. If so, it includes the canceled amount on the new event. This may cause the new event to be a negative amount. The assumption is that over time, earned credits will compensate for the negative event amount. For a membership that interfaces information to a third party, it is assumed that the negative event amount is also interfaced.

Service Credits Redeemed Through Billing

For some memberships, your customer may redeem their earned points by receiving a discount on their bill. For example, if your cable customer has earned one free pay-per-view movie, you can give them a credit the next time they get billed for a pay-per-view movie.

The base product has not provided any algorithms to credit a bill based on earned service credits. This section will identify considerations your implementers should follow when designing algorithms to redeem service credits through billing.

The recommendation is to credit the customer's bill by generating an adjustment using a pre-bill completion algorithm. This algorithm's responsibilities are as follows:

- Determine if the appropriate bill segment(s) contain the charges that need to be credited. For example, if the service credit is for a free pay-per-view movie, determine if the customer has been billed for a pay-per-view movie.
- Determine if the customer's current service credit balance for this program. For example, how many free pay-per-view movie credits are left?
- Create an adjustment to credit the appropriate service agreement by the eligible credit amount.
- Update the service credit membership balance by creating a new SC event with a credit for the number of points redeemed. Link the adjustment to the event as a contributed to FT.
- Consider cancel/rebill situations. If a cancel/rebill has occurred, determine if there is a change to the redeemed credits. For example, if the original bill had one pay-per-view movie that was credited and the new bill also has one pay-per-view movie, no change is needed to the service credit balance. If the original bill had more pay-per-view movies than the new bill, perhaps one or more redeemed service credit points should be reinstated. If the original bill had fewer pay-per-view movies than the new bill, perhaps more points should be redeemed.

You may wonder why we don't recommend crediting the customer's bill while generating the bill segment. For example, use pre-processing calculation rule to determine if any points should be redeemed and use a calculation rule to generate a bill calculation line with the credit amount.

The reason for this is that cancel/rebill logic is not straightforward. Algorithms executed during rate application should NOT perform any updates, such as updating the service credit membership balance. The balance should be updated using a bill segment freeze algorithm or a bill completion algorithm.

When a cancellation occurs, the service credit balance should be updated to reinstate the redeemed points. Again, this should occur when the cancellation is frozen or at bill completion time. If you perform a cancel/rebill, the calculation of the rebill segment does not have the up-to-date information about the service credit balance because the reinstatement of the points by the canceled segment has not occurred yet.

Designing Your Rate Options for Capital Credits

The capital credit allocation background process relies on certain data configuration in order to function correctly. This section identifies the required data setup.

●

FASTPATH:

Refer to Allocating Capital Credits for more information.

Identifying SQ and Sales Information for Historical Bill Segments

To allocate capital credits, the background process retrieves billing history for each SA that contributes to the membership for the given fiscal year. The process needs to calculate the service quantity (SQ) amount billed and the sales amount billed for the SA in that year.

NOTE:
Sales Amount. The sales amount refers to the monetary amount billed. For a capital credit allocation, this amount would generally exclude taxes and may exclude other line item amounts from the bill.

In order to calculate the amounts correctly, the background process must determine which bill calculation lines for each bill segment contain the SQ and/or sales information. Characteristics on the bill calculation lines identify which bill lines should be used.

The rest of this section uses examples to illustrate how you may configure your rate options to support this.

Define Characteristics for SQ and Sales

Char Type for identifying bill lines that hold **SQ information**

- Char Type: CCA-SQ
- Description: Capital credit allocation usage
- Type of Char: Pre-defined
- Values: Y
- Char Entities: Calculation Rule, Bill Calculation Line

Char Type for identifying bill lines that hold **sales information**

- Char Type: CCA-SALES
- Description: Capital credit allocation sales
- Type of Char: Pre-defined
- Values: Y
- Char Entities: Calculation Rule, Bill Calculation Line

Define Calculation Rules

Identify each rate schedule that may be linked to a service agreement that contributes to a capital credit membership.

For each of these rate schedules, identify the calculation rules whose resulting bill calculation line will contain a billable service quantity that should be included in the SQ calculation for allocating capital credits. For each calculation rule that qualifies, define the CCA-SQ char type (defined above) and a char value of Y.

For each rate schedule, identify the calculation rules whose resulting bill calculation line amount should be included in the sales calculation for allocating capital credits. For each calculation rule that qualifies, define the CCA-SALES char type (defined above) and a char value of Y.

➤ **NOTE:**
Characteristic Information. The system automatically copies characteristic info from a calculation rule to its resulting bill calculation line if the char type entities include both calculation rule and bill calc line.

Define Batch Control Parameters

The background process to allocate capital credits CPCRALOC receives the char type and char value to identify the bill calculation lines that contain the SQ and sales amounts. Once you have your characteristics defined, update your batch control to include these values as default parameter values.

Designing Bill Factors for Credit Allocation

The capital credit allocation process uses an allocation factor in its calculation. A typical capital credit membership may define multiple subcategories, meaning that allocation amounts are calculated each year for the multiple subcategories. The calculation is the same for each subcategory, but the allocation factor differs.

The process has been designed to calculate the allocation for a single subcategory. If your organization requires allocations calculated for multiple subcategories, the process must be run for each subcategory. The allocation process receives a bill factor as an input parameter. As a result, a different bill factor should be set up to define the allocation factor for each subcategory.

For each subcategory, the allocation factor may differ further for the type of customer. For example, the allocation for a commercial customer may differ from the allocation factor for a residential customer. The allocation background process expects the bill factor for each subcategory to define a characteristic type of revenue class. The process determines each service agreement's revenue class by looking at the value defined on its SA type.

Following is an example of bill factors set up for a capital credit membership with two subcategories: transportation and generation.

Characteristic Type for Allocation Bill Factor

Char Type: REV-CLASS

Description: Revenue Class

Type of Char: Pre-defined

Values: (define all the valid revenue class values)

Char Entity: Bill Factor

Bill Factor for Transportation

Bill Factor Id: CCAF-TRANS

Description: Transportation Allocation Factor

Char Type: REV-CLASS

Char Source: Characteristic Collection

Char Values: (for each year, the new transportation allocation factor for each revenue class must be defined)

Bill Factor for Generation

Bill Factor Id: CCAF-GEN

Description: Generation Allocation Factor

Char Type: REV-CLASS

Char Source: Characteristic Collection

Char Values: (for each year, the new generation allocation factor for each revenue class must be defined)

> **NOTE:**
> **The characteristic source is characteristic collection.** It is the responsibility of the background process to determine the SA's revenue class and to pass this value into the bill factor routines to retrieve the correct bill factor value.
>
> **Estimating Allocation Factors**. Often the company needs to estimate the allocation factors for the new fiscal year and may adjust the values several times until the calculated allocation amounts are satisfactory. Refer to Allocating Capital Credit for more information.

Partial Retirement

In the cooperative business, it is common to never retire certain capital credit allocation amounts. The amounts that do not retire should be assigned their own subcategory.

When executing the retirement background process, the subcategory to retire may be input to the process. If you have certain subcategories that you do not retire, you would simply run the background process for the subcategories that do retire.

Cooperatives typically retire amounts and transfer the amounts to a beneficiary when a member dies. This is known as "estate retirement". Refer to Service Credits for Capital Credit Memberships for more information. If your business practice designates that certain subcategories of allocated amounts do not get retired, this probably holds true for estate retirement as well. If that is the case, your membership inactivation algorithm should be designed to only retire the appropriate amounts by subcategory.

Interface Membership Information to a Third Party

For some memberships, you may accumulate points for a third party, for example accumulating frequent flier miles for an airline. For these types of memberships, you must interface the event information to the third party.

To interface information to a third party, you may choose one of the following options:

- Design an extract program to interface the information
- Use workflow and notification to interface the information via the XAI tool

Interface Via an Extract Program

The service credit event may indicate a batch code and batch run number. Design a program to extract event information to a third party. This extract program would select service credit events marked with its batch code and the current run number.

Your service credit event must define a completion algorithm that stamps the appropriate batch code and run number. The base product provides an algorithm type to perform this logic. Refer to SCEC-BT for more information.

Other Considerations For Interfacing Info to a Third Party

Because event information is extracted to a third party, you must consider how to handle adjustments to the event amount. For example, if your event is generated based on the creation of a bill segment, what should happen if that bill segment is cancelled?

You may want to prevent these types of events from getting canceled. Validation like this may be added via a user exit or using an event cancellation algorithm.

➤ **NOTE:**
Sample Algorithms. The product does not provide any base algorithms to prevent an event from being canceled.

You should allow negative event amounts to be created so that this information may also be sent to the third party's system.

Defining Loan Options

The topics in this section describe how to set up the system to enable loan functionality.

> **NOTE:**
> **Loans are optional.** The system configuration requirements described in this section are only relevant if your organization loans money to customers.

The Terms Of A Loan Are Stored On A Service Agreement

Loans are initiated by creating a loan service agreement for a customer. The loan service agreement (and its SA Type) contains the loan's terms:

- The **loan amount** is held in the service agreement's Total Amount to Bill.
- The **customer's periodic payment amount** is held in the service agreement's Recurring Charge Amount.
- The **number of amortization periods** (e.g., 36 months, 240 months, etc.) is held in the service agreement's Number of Payment Periods.
- If the **interest rate** is the same for all customers with this type of loan service agreement, the interest rate is defined on the service agreement's SA type (using a bill factor). A specific interest rate can be defaulted from a start option contract value. If a specific interest rate applies to the customer, the SA type's interest rate can be overridden by specifying a bill factor value on the customer's service agreement (where the bill factor value contains the specific interest rate for the customer).
- The SA type controls the **periodicity of the bills** (e.g., monthly or bi-weekly).

Because a loan is defined using a service agreement, the typical functionality that is controlled by the service agreement's SA type is supported, including:

- How and when it is billed.
- How payments are booked in the GL (and the payment priority relative to other service agreements).
- How its debt is monitored by credit and collections.
- How late payment charges are calculated.

Loan service agreements are created using Start/Stop just like all other service agreements. The start/stop transaction has special loan functionality that allows an operator to specify the service agreement-specific loan terms described above. A start option can be specified to override the SA type's interest bill factor.

> **NOTE:**
> **Automatic calculation of periodic payment amount / number of periods.** The system calculates a loan's periodic payment amount or number of payments (whichever is left blank). You can have the system do this on Start/Stop Maintenance (using the **Calculate** button that appears on the start service confirmation window), and on Loan - Main (by clicking the **Calculate** button). Regardless of where you do this, the calculation is

performed by an algorithm on the loan's SA type. Refer to the LPDA-SI algorithm type for more information about the base package algorithm.

Payoff Balance and Current Balance for Loans

As described under Current Amount versus Payoff Amount, a loan service agreement's current balance and payoff balance differ during the lifetime of the loan. Current balance contains how much the customer owes to remain current (typically their periodic payment amount), and payoff balance contains the amount the customer would have to pay to payoff the loan (typically the principal balance plus any accrued interest charges).

Unlike other SAs, loans have two accounts receivable distribution codes: long term and short term. These two codes allow the general ledger to differentiate between unbilled loan receivables (long term) and billed loan receivables (short term). The current balance for a loan is always the amount of the short-term receivables. The payoff balance for a loan is always the net of the short-term receivables and the long-term receivables.

If the SA has a special role of Loan, the financial transaction algorithms supplied with the base package transfer the current amount between the long-term receivables and the short-term receivables in the GL. For example, when a bill segment is generated for the loan SA, the amount of the periodic payment is transferred from the long-term receivables to the short-term receivables. Don't worry, the examples in the following sections show exactly what these transactions look like.

An operator can see the how the bills, payments and adjustments have affected the GL, current balance and payoff balance using SA Financial History.

The following sections provide examples of how adjustments, bills and payments are recorded in the GL and the subsequent effect on the current and payoff balances. When reading the examples, remember that the payoff balance is always the net of the short-term receivable and the long-term receivable balances.

Booking The Principal Amount Using An Adjustment

When a loan service agreement is activated (i.e., when its status changes from pending start to active), an adjustment is created to book the principal amount. If the customer takes out a loan of 10,000, the adjustment's financial transaction looks as follows:

Event	GL Accounting	Effect On Current Balance	Effect On Payoff Balance	Current Balance	Payoff Balance
Loan service agreement is activated (and an adjustment is created to book the principal)	Long Term Loan Receivable 10,000 Cash <10,000>	0	10,000	0	10,000

This adjustment is issued if:

- The service agreement's SA type indicates a special role of Loan.
- The loan service agreement's Total Amount to Bill contains an amount (i.e., the loan amount).
- The loan service agreement was created using a start option on Start/Stop Maintenance AND the start option references an adjustment type and this adjustment type has been set up as follows:

- The adjustment type's distribution code should reference the GL account to credit (e.g., Cash).
- The adjustment type's FT algorithm reference Payoff Amt = Adj Amt / Current Amt = 0 (booking principal only impacts a customer's payoff balance).

Note that because this financial transaction doesn't have a current amount (the customer doesn't actually owe a current amount yet), there is no need to book anything to the short-term receivables distribution code.

Loan Amortization Schedule Calculation

The amortization schedule is a projection of the amount of principal and interest in each payment over the life of the loan. The amortization schedule may change, for example if the interest rate changes or the customer makes an overpayment (reducing the principal balance).

A loan's amortization schedule is calculated when an operator clicks the **Calculate** button on Loan - Main. Please be aware that when this button is clicked, an algorithm plugged in on the loan SA type actually calculates the amortization schedule (refer to the algorithm type LSCH-SI for more information).

Billing For Loans And Interest Calculation

A bill segment is produced for a loan when its service agreement's account is next billed.

Factors on a loan's SA type controls when a bill segment is produced for a loan. Typically SA types for loan service agreements are set up to use anniversary calendar billing. For this configuration:

- You must indicate the type of anniversary billing in the calendar billing option. Currently, we only support the Anniversary Future Billing (meaning that loans are billed in advance just like a classic home loan is). Refer to Using The Anniversary Method for more information about how this billing method controls the end date of the loan bill segment.
- You must reference an anniversary billing frequency consistent with the recurring charge frequency (e.g., monthly, quarterly, etc.).

To set up for a loan that is billed on a monthly basis, you would define the following fields in SA Type - Billing:

- **Use Calendar Billing**: Anniversary Future Billing
- **Anniversary Bill Frequency**: Monthly
- **Total Bill Amount**: Required
- **Recurring Charge**: Required
- **Recurring Charge Frequency**: Monthly
- **Total Amount To Bill Label**: Loan Amount
- **Recurring Chg Amt Label**: Payment Amount

If your type of loan must be billed on an exact date (for example, always on the 15th of the month) or with an exact number of days between each bill (for example, every 14 days), then your loan should be set up to use the calendar billing option of Use Bill Period. In order for your loan bills to be created on specific dates, the system makes the following assumptions:

- Your bill cycle schedule for the loan's account is defined with the dates that you want the loan to bill and is defined with the window start date equal to the window end date.
- You define a bill period schedule corresponding to your bill cycle schedule. Each bill period schedule's bill date should match your bill cycle window start date and each bill period schedule's bill segment end date should be set to the loan period end date.

- Considerations for the first bill. If the loan SA Type's Initial Start Date Option indicates that the first day of the service agreement should be billed, then the loan SA's start date should match the window start date of the next bill cycle schedule for the account. If the loan SA Type's Initial Start Date Option indicates that first day of the service agreement should not be included, then the loan SA's start date should be one day prior to the window start date of the next bill cycle schedule for the account.
- Define your recurring charge frequency to match the frequency of your bill periods.

How the bill segment affects the customer's balance, and how it affects the general ledger are controlled by several algorithms defined on the loan service agreement's SA type and bill segment type:

- The SA type's loan schedule algorithm controls how the loan amortization schedule is calculated.
- The SA type's loan interest charge algorithm controls how interest is calculated.
- The bill segment type's create algorithm controls how the bill lines are constructed.
- The bill segment type's financial algorithm controls how the general ledger is affected by the bill.

The second entry in the following table contains an example of the financial transaction produced by the base package algorithms (note, the first financial transaction in the table was described under Booking The Principal Amount Using An Adjustment).

Event	GL Accounting	Effect On Current Balance	Effect On Payoff Balance	Current Balance	Payoff Balance
Loan service agreement is activated	Long Term Loan Receivable 10000 Cash <10000>	0	10,000	0	10,000 long-term: 10,000
First bill segment is produced	Interest: Long Term Loan Receivable 41.66 Interest Revenue <41.66> Transfer Long Term To Short Term: Short Term Loan Receivable 438.71 Long Term Loan Receivable <438.71>	438.71	41.66	438.71	10,041.66 short-term: 438.71 long-term: 9,602.95

Several important points are illustrated above:

- When a bill segment is produced for a loan service agreement, the following takes place (if you use a bill segment creation algorithm of Create Bill Segment for Loans):
  - The loan SA type's bill segment creation algorithm calls the SA type's loan interest charge algorithm.
  - The base package loan interest charge algorithm calculates simple interest based on: 1) unbilled principal (i.e., the service agreement's payoff balance minus the current balance), 2) the number of billing periods covered by the bill, and 3) the interest rate. Refer to the algorithm type LINT-SI for more information about the base package interest calculation algorithm.

> **NOTE:**
> **No interest on interest and no interest on past due amounts.** Just like a classic home loan, the base package algorithms do not calculate interest on interest, nor do they calculate interest on past due amounts.

> If you want to levy a late payment charge, use the SA type's late payment processing. If your organization calculates interest differently, you must develop your own algorithm(s).

- The SA type's bill segment creation algorithm uses the calculated interest to format the bill segment's bill lines. It creates one bill line to show the amount of interest in the payment, and another bill line to show the amount of principal. The principal amount is equal to the service agreement's periodic payment amount minus the amount of calculated interest.
- The financial transaction illustrated above is created if you use a bill segment financial algorithm of Payoff Amt = Interest / Current Amt = Bill Amount on the loan's bill segment type. The following explains how this algorithm works:
  - The SA's current balance increases by the amount of the loan's periodic payment amount (i.e., its recurring charge amount). In other words, the amount the customer thinks they owe increases by 438.71.
  - The SA's payoff balance increases by the amount of interest. In other words, if the customer wanted to payoff the loan, they'd owe 10,041.66.
  - The *Interest* portion of the GL accounting is straightforward (if you're an accountant). It simply takes the amount of interest and debits it to the SA type's receivable distribution code (long-term receivables) and credits it to the distribution code defined on the interest rate's bill factor (typically interest revenue).
  - The *Transfer* portion of the GL accounting transfers moneys from long-term receivables (i.e., the unbilled principal) to short term receivables (the billed portion of the debt). The amount transferred is equal to the FT's effect on the service agreement's current balance, allowing the general ledger to differentiate between unbilled loan receivables (long term) and billed loan receivables (short term). Remember that the payoff balance is the net of the short-term and long-term receivables.

Paying What Is Owed

When a payment is made for a loan:

- The service agreement's payoff amount is reduced by the payment amount.
- The service agreement's current amount is reduced by the payment amount.

The events that happen when a customer makes a payment against a loan is controlled by the FT algorithm plugged in on the loan service agreement's payment segment type. We'll use an example to help explain how this algorithm works. The 3 rd entry in the following table illustrates the financial transaction produced when a payment is made (note, the first two financial transactions were described above).

Event	GL Accounting	Effect On Current Balance	Effect On Payoff Balance	Current Balance	Payoff Balance
Loan service agreement is activated	Long Term Loan Receivable 10000 Cash <10000>	0	10,000	0	10,000 long-term: 10,000
First bill segment is produced	Interest: Long Term Loan Receivable 41.66 Interest Revenue <41.66> Transfer Long Term To Short Term:	438.71	41.66	438.71	10,041.66 short-term: 438.71 long-term: 9,602.95

	Short Term Loan Receivable 438.71 Long Term Loan Receivable <438.71>				
Payment is made	Affect Cash Cash 438.71 Long Term Loan Receivable <438.71> Transfer Long Term To Short Term: Long Term Loan Receivable 438.71 Short Term Loan Receivable <438.71>	-438.71	-438.71	0	9,602.95 short-term: 0 long-term: 9,602.95

The financial transaction illustrated above is created if you use a payment segment FT creation algorithm of Payoff Amt = Current Amt = Pay Amt on the loan's payment segment type. The following explains how this algorithm works:

- The SA's current balance decreases by the amount of the payment amount. In other words, the customer thinks they owe 0 after the payment. Note refer to Overpayments for information about how an overpayment affects the SA's balances and the general ledger.
- The SA's payoff balance decreases by the amount of the payment. In other words, if the customer wanted to payoff their loan, they'd owe 9,602.95.
- The *Cash* portion of the GL accounting is straightforward (if you're an accountant). It simply takes the amount of the payment and debits it to the payment segment type's distribution code (typically cash) and credits it to the SA type's receivable distribution code (long-term receivable).
- The *Transfer* portion of the GL accounting effectively reduces short-term receivables by the FT's effect on the customer's current balance. This reduction is handled by an offsetting increase to long-term receivables (to make up for reduction made when the cash was applied). Again, this differentiation between short-term and long-term receivables allows the general ledger to differentiate between unbilled loan receivables (long term) and billed loan receivables (short term).

Overpayments On Loans

You can determine whether you want to accept loan overpayments. Overpayments reduce the principal amount (the amount owed on the loan), which follows the philosophy adopted by a typical home loan.

When the payment is made, any overpayments are distributed according to the overpayment distribution algorithm defined for the customer class. Any customer classes for which you want to allow loan overpayments should use an overpayment distribution algorithm that keeps the overpayment on the loan SA.

When the payment transaction is frozen, the system checks to see if there is a credit amount on the loan SA's current balance. If a credit exists, the customer has made an overpayment and an adjustment is created to zero out the current balance and transfer the amount of the credit from the SA's short-term receivable to long-term receivable. The adjustment may appear on the customer's next bill to show the additional amount paid against the principal.

The algorithm that controls this adjustment to remove the credit on current balance is plugged in on the loan service agreement's SA type and is applied on the SA Type - Payment Freeze system event.

The third and fourth entries in the following table illustrate an overpayment (note, the first two financial transactions were described above).

Event	GL Accounting	Effect On Current Balance	Effect On Payoff Balance	Current Balance	Payoff Balance
Loan service agreement is activated	Long Term Loan Receivable 10000 Cash <10000>	0	10,000	0	10,000 long-term: 10,000
First bill segment is produced	Interest: Long Term Loan Receivable 41.66 Interest Revenue <41.66> Transfer Long Term To Short Term: Short Term Loan Receivable 438.71 Long Term Loan Receivable <438.71>	438.71	41.66	438.71	10,041.66 short-term: 438.71 long-term: 9,602.95
Payment is made (with an overpayment of 200.00)	Affect Cash Cash 638.71 Long Term Loan Receivable <638.71> Transfer Long Term To Short Term: Long Term Loan Receivable 638.71 Short Term Loan Receivable <638.71>	-638.71	-638.71	-200.00	9,402.95 short-term: <200.00> long-term: 9,602.95
Create adjustment to remove SA's credit.	Transfer Short Term Credit to Long Term: Short Term Loan Receivable 200.00 Long Term Loan Receivable <200.00>	200.00	0	0	9,402.95 short-term: 0 long-term: 9,402.95

In the third financial transaction illustrated above, the billed amount of the payment works essentially the same as that illustrated under Paying What Is Owed. If the Keep Overpayment on Loan SA algorithm is plugged in on the overpayment distribution event on the customer class, the overpayment amount is applied to the loan SA, creating a credit on the SA's current balance. The following explains how this algorithm works:

- If the account has an loan SA and there is an excess credit, the credit is applied to the loan SA (as long as this does not cause the loan SA to have a credit payoff balance).
- If there is not a loan SA to which the credit can be applied, the algorithm checks to see if there is an open excess credit SA for the account.
  - If so, the excess credit amount is applied to the excess credit SA.
  - If not, the algorithm creates an excess credit SA and applies the amount to this SA.

The fourth financial transaction illustrated above is created if the Create Adjustment to Remove SA's Credit algorithm is plugged in on the loan's SA type's payment freeze event. The following explains how this algorithm works:

- If the SA's current balance is less than zero, the algorithm creates a frozen adjustment that removes the credit by transferring the credit from the short-term receivable to the long-term receivable. This adjustment ID is captured on the pay segment so the adjustment can be canceled if the payment is later canceled. Note that the adjustment cancel reason used by the system in this case is specified on the Financial Transaction Options Feature Configuration using the Adjustment Cancel Reason For Payment Linked To Adjustment option type.
- The SA's current balance increases by the amount of the credit transfer. In other words, the customer thinks they owe 0 after the transfer.
- The SA's payoff balance doesn't change because the payoff balance is always the net of the short-term and long-term receivables. In other words, if the customer wanted to payoff their loan, they'd still owe 9,402.95.

➤ **NOTE:**
Overpayments and interest. The base package interest calculation algorithm (plugged in on the loan's SA type) does not take into consideration the exact date that the overpayment is made when calculating the interest for the period. It only takes into consideration the outstanding principal amount (payoff balance - current balance) at the time of the interest calculation.

Adjusting Loan Amounts

You would issue ad hoc adjustments if you need to change a loan's payoff and/or current balance outside of the normal billing / payment functions.

An adjustment can:

- Reduce a loan's payoff balance.
- Reduce a loan's current balance (i.e., how much the customer thinks they currently owe).
- Reduce both the loan's payoff and current balance.

For adjustments that affect payoff amount only:

- These adjustments are used to change the principal owed, e.g., if an additional amount is loaned.
- The adjustment's adjustment type should reference the Payoff Amt = Adj / Current Amt = 0 FT algorithm.
- GL lines will be generated to reflect the change to principal.

For adjustments that reduce current amount only:

- These adjustments can be used to change the amount that the customer must pay as part of the next bill.
- The adjustment's adjustment type should reference the Payoff Amt = 0 / Current Amt = Adj Amount (no GL) FT algorithm.
- Typically, GL lines are not generated for FTs that only affect the customer's current balance. However, moneys must be transferred from long to short in the amount of the adjustment (as described above under Payoff Balance and Current Balance for Loans).

For adjustments that reduce both payoff and current amount:

- These adjustment types can be used to levy additional charges, such as late fees, or to correct interest calculations.
- The adjustment's adjustment type should reference the Payoff Amt = Adj / Current Amt = Adj FT algorithm.
- GL lines are generated to reflect the change to principal. In addition, GL lines must be generated to transfer money from long to short in the amount of the adjustment (as described above under Payoff Balance and Current Balance for Loans).

➤ **NOTE:**

Adjustments that affect the principal balance (payoff balance - current balance) affect the term of the loan because interest is based on the principal balance.

Adjustments can cause credit balances to exist. If you credit the loan SA, it is possible for the current balance to become negative. You may need to create additional adjustments that affect the current amount, depending on whether the customer needs to pay this amount as part of the next payment.

Writing Off Loans

Loans are written off using the standard write-off processing.

FASTPATH:
Refer to The Big Picture Of Write Off Processing for background information.

We illustrate the classic financial transactions that transpire to financially write-off a loan to help illustrate important points (these are the fourth and fifth entries in the following table):

Event	GL Accounting	Effect On Current Balance	Effect On Payoff Balance	Current Balance	Payoff Balance
Loan service agreement is activated	Long Term Loan Receivable 1000 Cash <1000>	0	10,000	0	10,000 long-term: 10,000
First bill segment is produced	Interest: Long Term Loan Receivable 41.66 Interest Revenue <41.66> Transfer Long Term To Short Term: Short Term Loan Receivable 438.71 Long Term Loan Receivable <438.71>	438.71	41.66	438.71	10,041.66 short-term: 438.71 long-term: 9,602.95
Payment is made (with an overpayment of 200.00)	Affect Cash Cash 638.71 Long Term Loan Receivable <638.71> Transfer Long Term To Short Term: Long Term Loan Receivable 638.71 Short Term Loan Receivable <638.71>	-638.71	-638.71	-200.00	9,402.95 short-term: <200.00> long-term: 9,602.95

Create adjustment to remove SA's credit.	Transfer Short Term Credit to Long Term: Short Term Loan Receivable 200.00 Long Term Loan Receivable <200.00>	200.00	0	0	9,402.95 short-term: 0 long-term: 9,402.95
Sync up current with payoff balance	Transfer Long Term To Short Term: Short Term Loan Receivable 9,402.95 Long Term Loan Receivable <9,402.95>	9,402.95	0	9,402.95	9,402.95 short-term: 9,402.95 long-term: 0
Transfer balance to a write-off SA	Transfer From Loan SA: Write Off Xfer Clearing 9,402.95 Long Term Loan Receivable <9,402.95> Long Term Loan Receivable 9,402.95 Short Term Loan Receivable <9,402.95> Transfer To Write Off SA: Bad Loans Expense 9,402.95 Write Off Xfer Clearing <9,402.95> Note, these will cause a balance of 9,402.55 to exist on the write-off service agreement.	-9,402.95	-9,402.95	0	0

The only unusual portion of the last two financial transactions is the impact on short and long term receivables. Please see the examples above under Billing For Loans And Interest Calculation and Paying What Is Owed to understand how any impact to a loan's current balance causes this type of financial transfer to occur.

Distribution Codes for Loans

As explained above under Payoff Balance and Current Balance for Loans, loans have two accounts receivable distribution codes: long term and short term. These two codes allow the general ledger to differentiate between unbilled loan receivables (long term) and billed loan receivables (short term). Both receivables distribution codes are defined on the loan SA type.

In addition, loans have a distribution code used to book interest revenue. The interest revenue distribution code is defined on the loan's bill factor value for a revenue class (defined on the loan's SA type). For example, on the bill factor you can use one distribution code to book interest revenue from the residential revenue class and another distribution code to book interest revenue from the commercial revenue class. In this example, you create two loan SA types, one for residential revenue and the other for commercial revenue.

Loans may also have a bad loan debt (expense) distribution code that is used when writing off a loan. The bad loan debt distribution code is defined on the loan's write-off service agreement type. Refer to Defining Credit & Collections Options for more information.

Setting Up The System To Enable Loans

The above topics provided background information about how loans are supported in the system. The following discussion summarizes the various setup tasks alluded to above.

Distribution Code

You must set up the following distribution codes:

- Long term receivables
- Short term receivables
- Interest revenue
- Bad loan debt

Adjustment Types

The following adjustment types are needed:

- Activate a loan. This should reference a distribution code associated with cash or the cash equivalent and an FT algorithm of Payoff Amt = Adj / Current Amt = 0.

> **NOTE:**
> **Creating Checks for Loan Amounts.** If you want the system to initiate a check to the customer for the loan amount, the loan activation adjustment type should indicate an **A/P Request Type Code**. Refer to Controls The Interface To A/P & 1099 Reporting for more information.

- Remove Credit (Overpayment). This adjustment should reference an FT algorithm of Payoff Amt = 0 / Current Amt = Adj Amount (no GL). Even though this algorithm indicates that there is no effect on the GL, there is a special exception built in for loan SAs. The special exception creates GL details to transfer the current balance (short-term receivable) to the long-term receivable. Refer to Overpayments for more information.

> **NOTE:**

> **Printing Overpayments on Bills**. If you want the overpayment adjustment to appear on customers' bills, turn on **Print by Default** and enter a **Description on Bill** (e.g. "Additional Principal"). For more information, refer to Controls Information Printed On The Bill.

- Adjustment types to perform any of the adjustments described under Adjusting Loan Amounts.

Adjustment Type Profile

Create an adjustment type profile that references the adjustment types used on a loan. Besides the above adjustment types, it should also reference adjustment types to levy late payment charges (if applicable), levy non-sufficient funds charges (if applicable), refund overpayments (if applicable), sync current balance with payoff balance at write-off time, transfer balances to a write-off service agreement, and write-down small balances.

Algorithms

Add the following algorithms:

- Overpayment Distribution. This algorithm is later plugged in on the customer class for the Overpayment Distribution system event. Refer to the algorithm type OVPY-LO-CSA for more information about the base package algorithm.
- Bill Segment Creation for Loans. This algorithm is later plugged in on the loan's bill segment type. Refer to the algorithm type BSBS-LO for more information about the base package algorithm.
- Bill Segment Financial Transaction Creation for Loans. This algorithm is later plugged in on the loan's bill segment type. Refer to the algorithm type BSBF-LO for more information about the base package algorithm.
- Amortization Schedule. This algorithm is later plugged in on the SA type for the Loan Schedule system event. Refer to the algorithm typeLSCH-SI for more information about the base package algorithm.
- Interest Calculation. This algorithm is later plugged in on the SA type for the Loan Interest Charge system event. Refer to the LINT-SI algorithm type for more information about the base package algorithm.
- Payment Periods/Payment Amount Calculation. This algorithm is later plugged in on the SA type for the Loan Periods and Amount system event. Refer to the algorithm typeLPDA-SI for more information about the base package algorithm.
- Loan SA Payment Freeze. This algorithm has a parameter that must reference the Remove Credit adjustment type defined above. This algorithm is later plugged in on the SA type for the Payment Freeze system event. Refer to thealgorithm type STPZ-RMVCR for more information about the base package algorithm.

Bill Factor

You must set up a bill factor that defines the interest rate. If you have different interest rates for different types of loans, you can create a separate bill factor for each or you can use start options to override the interest rate.

On the bill factor's bill factor value, make sure to reference the GL distribution code used to book interest revenue for the revenue class specified on the loan's SA type.

Customer Class

Any customer class on which you want to allow overpayments for a loan must use the overpayment distribution algorithm defined above. The overpayment distribution algorithm keeps the overpayment on the loan SA rather than transferring it to an excess credit SA, allowing a subsequent adjustment to apply the overpayment to the principal balance.

Bill Segment Type

Create a bill segment type that references the bill segment creation algorithm defined above and the FT creation algorithm defined above.

Frequency

Create frequency codes to correspond to the frequency of your loans. When setting up your SA types, you must indicate a recurring charge frequency and, if you use the Anniversary Billing Option, you must indicate an anniversary frequency.

Bill Period

If you use the Use Bill Period option, set up a bill period with an appropriate schedule of dates for billing your loan.

Bill Cycle Schedule

If you use the Use Bill Period option, the bill cycle schedule for these types of loans should be defined with an appropriate schedule of dates for billing your loan.

Collection, Severance and Write Off Processes

You should set up the appropriate credit and collections information. Refer to Defining Credit & Collections Options for more information.

SA Type

You must set up a SA type for your loan service agreements (you may need multiple SA types if you have different business rules for different types of loans). The following points describe the minimal requirements for a loan SA type.

SA Type - Main

Define the following options:

- **Distribution Code** should be a long-term receivable code.
- **Service Type** should references something like "Miscellaneous Service".
- Specify the **Revenue Class** that, together with the interest bill factor, determines the distribution code used to book the loan interest revenue.
- **Start Option** should be turned on.
- The Payment Segment Type should reference the Normal Payment. The base package payment segment financial algorithm (Payoff Amt = Current Amt = Pay Amt) used for Normal Payment pay segment types creates the additional GL details to transfer the credit from long-term receivables to short-term receivables if the SA's special role is Loan.
- Turn on **Late Payment Charge** if applicable.
- Define an appropriate **Adj**ustment **Type (Synch Current)** that will cause current balance to be synchronized with payoff balance (if the loan is written off).

SA Type - Detail

Define the following options:

- **Special Role** is Loan.
- Specify the **Interest Bill Factor** set up above. You can use start options to override.
- Use **Override Interest Flag** to indicate whether the interest rate defined on the interest bill factor may be overridden at the SA level. If you select Allowed, the interest rate may be overridden by a contract value on a start option or the SA.
- Use the short-term receivable account defined above as the **Loan A/R Distribution Code**. If you do not want the system to differentiate between short-term receivables and long-term receivables, make the loan A/R distribution code the same as the distribution code (above).

SA Type - Billing

- **FASTPATH:**
  For an overview of some of these options, be sure to refer to Billing For Loans And Interest Calculation for more information.

Define the following options:

- The Bill Segment Type should reference the value created above.
- **Characteristic Premise Req**uired should not be checked. (A loan SA is not a premise-based service. SA types that have this box checked are filtered out of the SA type search when the start method is Start a SA, so users will never be able to start a loan SA that requires a characteristic premise.)
- **Use Calendar Billing** must equal Anniversary Future Billing or Use Bill Period.

- **Bill Period** is required for the Use Bill Period option.
- **Anniversary Billing Frequency** is required for the Anniversary Future Billing option and must equal the periodicity of the bills (monthly, weekly, etc.).
- **Total Bill Amount** is required (it holds the principal).
- **Total Amount To Bill Label** should reference something like "Loan Amount".
- **Recurring Charge** is required (it holds the periodic payment amount).
- **Recurring Chg Amt Label** should reference something like "Payment Amount".
- **Recurring Charge Frequency** is required (it defines the periodicity associated with the recurring charge amount) and must be the same as the bill period frequency or the anniversary billing frequency (based on your **Use Calendar Billing** option).

SA Type - Rate

Turn off the **Rate Required** switch as loans do not use rates.

SA Type - SP Type

Turn off the **Service Points Required** switch as loans do not have service points.

SA Type - Adjustment Profile

Adjustment Type Profile should reference the profile set up above.

SA Type - Credit and Collections

The credit and collections information should reference a **Severance Process Template** that simply expires the loan. The **Debt Class** and **Write Off Debt Class** should reference an appropriate value consistent with your credit and collections rules. Refer to Defining Credit & Collections Options for more information.

SA Type - Algorithms

The Loan Schedule, Loan Interest Charge, Loan Periods, and Amounts and Payment Freeze algorithms defined above must be set up.

Start Options

Loans require a start option to define the adjustment type that is used to create the adjustment to book the initial principal amount when the loan is activated. Loan start options can also specify default values for loan amount, payment amount, and number of periods.

Create at least one start option for each loan SA type. The following information should be defined:

- **Adjustment Type** should reference the adjustment type defined above to book principal.
- **Recurring Charge Amount** (Payment Amount) should only be defined if you have standard loan payments.
- **Total Amount To Bill** (Loan Amount) should only be defined if you have standard loan amounts.
- **Number of Payment Periods** should be the number of payment periods in the loan (if you have a standard loan period).
- If you want to define a set of standard APRs for a loan SA Type, set up a **Contract Value** on the Rate Info page. The **Bill Factor** should match the interest bill factor specified on the SA Type.

Defining Non-Billed Budget Options

A Non-Billed Budget (NBB) is a payment plan that allows your customers to pay set amounts at specified intervals (e.g. every two weeks). Non-Billed Budgets are typically used when your company bills on an infrequent basis and you want to provide your customers with a mechanism to make smaller payments more frequently. As the name suggests, bills are not created for the Non-Billed Budget's scheduled payments; customers must remember to make their payments at the scheduled intervals. The topics in this section describe how to design and set up non-billed budgets.

➤ **NOTE:**
Non-Billed Budgets are optional. The system configuration requirements described in this section are only relevant if your organization offers Non-Billed Budgets.

What Is A Non-Billed Budget?

A Non-Billed Budget is a special type of budget or payment plan that encompasses three major elements:

- A set of scheduled payments
- The business rules used to recommend and potentially renew the payment schedule
- The business rules that govern the financial impact on the current and payoff balances of the SAs covered by the payment schedule

Non-Billed Budgets are managed via a service agreement whose SA type has a special role of Non-Billed Budget. If an SA type has a special role of Non-Billed Budget it must have one or more recommendation rules, and SAs of that type are allowed to have payment schedules.

The Financial Impact Of Non-Billed Budgets

When you set up the Non-Billed Budget SA type, you can indicate whether the Non-Billed Budget is monitored by the account debt monitor process. The following sections contain examples of financial transactions for Non-Billed Budgets that are monitored by the account debt monitor.

➤ **NOTE:**
Unmonitored Non-Billed Budgets. SAs that are covered by unmonitored Non-Billed Budgets are subject to different financial treatment than those that are monitored. The financial transactions relevant for unmonitored Non-Billed Budgets are discussed later.

Current Amount For SAs Covered By A Non-Billed Budget

As described under Current Amount versus Payoff Amount, the values of the current balance and payoff balance are the same for most financial transactions. One exception is for SAs that are covered by a Non-Billed Budget in which case the current balance is always zero and the payoff balance is always the amount the customer really owes.

When a Non-Billed Budget is activated or an SA is added to a Non-Billed Budget, the system creates adjustments for all affected SAs to set the current amount equal to zero. The adjustment type that references the Payoff Amt = 0 / Current Amt = Adj Amount (no GL) algorithm is taken from the SA's SA type.

> **NOTE:**
> **Non-Billed Budgets and Open Item Accounts.** If the Non-Billed Budget is for an open item account, no match event is created for the financial transaction that reduces the current amount to zero. This FT must be manually matched if required.

The following example shows the effect of activating a Non-Billed Budget that covers an electric SA with a current balance of 25. The transactions for the electric SA are illustrated on the right and the transactions for the non-billed budget SA are illustrated on the left.

Event	Effect On Current Balance	Effect On Payoff Balance	Current Balance	Payoff Balance	Effect On Current Balance	Effect On Payoff Balance	Current Balance	Payoff Balance
	Electric SA				Non-Billed Budget SA			
Starting balance	0	0	25	25				
Non-Billed Budget is activated	-25	0	0	25				

By setting the current balance for the covered SAs to zero, the SAs are insulated from the regular debt monitoring process. Depending on the Non-Billed Budget recommendation algorithm, the payoff balance can be ignored, divided evenly between the scheduled payments or added to the first scheduled payment.

Activating a Non-Billed Budget has no affect on its own current or payoff balances.

Scheduled And Actual Payments On The Non-Billed Budget

When a scheduled payment is due, an adjustment is created to increase the Non-Billed Budget's current balance by the expected amount. The current balance on the SA can be monitored to ensure that payments are being made on time.

When the payment is made, the Non-Billed Budget's current balance is reduced to zero and the Non-Billed Budget's payoff balance reflects the accumulated credit from the payment. This accumulated credit is transferred to the covered SAs when the next bill for the account is completed.

The following example illustrates scheduled and actual payments for the Non-Billed Budget in the previous example. The amount of the scheduled Non-Billed Budget payments is 10. Note that the first two transactions were described above.

Event	Effect On Current Balance	Effect On Payoff Balance	Current Balance	Payoff Balance	Effect On Current Balance	Effect On Payoff Balance	Current Balance	Payoff Balance
	Electric SA				Non-Billed Budget SA			
Starting balance	0	0	25	25				
Non-Billed Budget is activated	-25	0	0	25				
First scheduled payment is due					10	0	10	0
Payment					-10	-10	0	-10

An algorithm (Process NBB Scheduled Payment) plugged in on the Non-Billed Budget SA type creates the appropriate financial transactions when the scheduled payment is due. The algorithm is called by the Non-Billed Budget Scheduled Payment Processing (NBBPS) background process.

The normal payment processing handles the adjustments created when the payment is made.

NOTE:
Non-Billed Budget Payment Cancellation. If a payment is canceled, the financial transaction is reversed.

Automatic Payments And Non-Billed Budgets. Users may set up a Non-Billed Budget to use automatic payments by setting up the account's auto pay options. Users may also exclude the Non-Billed Budget from automatic payment if the account is set up for automatic payment. The NBB Scheduled Payment Automatic Payment Create background process calls the Auto Pay Creation algorithm to create the auto payment for a Non-Billed Budget.

Overpayments for Non-Billed Budgets

Typically, payments in excess of the Non-Billed Budget's current balance are credited to an overpayment (excess credit) SA. When the next adjustment is created for a scheduled payment, the credit on the overpayment SA is used to relieve the non-billed budget current balance.

NOTE:
Overpayment Distribution Algorithm. The overpayment distribution is a function of the overpayment distribution algorithm plugged in on the account's customer class. We strongly recommend that the Non-Billed Budget SA type be set up so that overpayment is not allowed. Any excess payments should go to an overpayment SA. For more information about overpayment distribution, refer to Overpayment Segmentation.

In the example below, the customer pays 20 for a scheduled payment instead of 10. The non-billed budget SA is illustrated on the right and the overpayment SA is illustrated on the left. The electric SA is not illustrated in the example below because scheduled payments, payments and overpayments have no effect on covered SAs. The first four transactions were illustrated above.

Event	Effect On Current Balance	Effect On Payoff Balance	Current Balance	Payoff Balance	Effect On Current Balance	Effect On Payoff Balance	Current Balance	Payoff Balance
	Non-Billed Budget SA				Overpayment SA			
Starting balance								
Non-Billed Budget is activated								
First scheduled payment is due	10	0	10	0				
Payment	-10	-10	0	-10				
Second scheduled payment is due	10	0	10	-10				
Over-payment	-10	-10	0	-20	-10	-10	-10	-10
Third scheduled payment is due	10	0	10	-20				
Transfer Adjustment	-10	-10	0	-30	10	10	0	0

After the Process NBB Scheduled Payment algorithm creates the next scheduled payment, it looks for a credit amount on the overpayment SA and creates an adjustment to transfer the credit balance (or the amount of the payment if the credit is more than the scheduled payment amount) from the overpayment SA to the non-billed budget SA. The overpayment transfer adjustment type and the overpayment SA type are specified as parameters to the algorithm.

Underpayments For Non-Billed Budgets

An insufficient payment or a canceled payment leaves a current balance on the Non-Billed Budget SA.

In the example below, the customer makes a payment of 7 for the fourth scheduled payment. The example below does not show the overpayment SA (illustrated above). The first eight transactions are discussed above.

Event	Effect On Current Balance	Effect On Payoff Balance	Current Balance	Payoff Balance	Effect On Current Balance	Effect On Payoff Balance	Current Balance	Payoff Balance
	Electric SA				Non-Billed Budget SA			
Starting balance	0	0	25	25				

Non-Billed Budget is activated	-25	0	0	25				
First scheduled payment is due					10	0	10	0
Payment					-10	-10	0	-10
Second scheduled payment is due					10	0	10	-10
Over-payment					-10	-10	0	-20
Third scheduled payment is due					10	0	10	-20
Transfer Adjustment					-10	-10	0	-30
Fourth scheduled payment					10	0	10	-30
Underpayment					-7	-7	3	-37

The Process NBB Scheduled Payment algorithm creates a trigger to ensure that the current balance is monitored by the account debt monitor. Refer to Credit And Collections And Non-Billed Budgets for more information.

Billing For SAs Covered By The Non-Billed Budget

When the next bill for the account is completed, the credit on the Non-Billed Budget is transferred to the covered SAs. The credit is prorated over the covered SAs according to the relative payoff balances on each SA.

In the example below, the electric SA's bill is 33. The first ten transactions are discussed above.

Event	Effect On Current Balance	Effect On Payoff Balance	Current Balance	Payoff Balance	Effect On Current Balance	Effect On Payoff Balance	Current Balance	Payoff Balance
	Electric SA				Non-Billed Budget SA			
Starting balance	0	0	25	25				
Non-Billed Budget is activated	-25	0	0	25				
First scheduled payment is due					10	0	10	0

Payment					-10	-10	0	-10
Second scheduled payment is due					10	0	10	-10
Over-payment					-10	-10	0	-20
Third scheduled payment is due					10	0	10	-20
Transfer Adjustment					-10	-10	0	-30
Fourth scheduled payment					10	0	10	-30
Underpayment					-7	-7	3	-37
Bill	0	33	0	58				
Bill completion (transfer adjustment)	0	-37	0	21	0	37	3	0

When a bill segment financial transaction is created, the current amount is set to zero for SAs that are covered by the Non-Billed Budget (if you use a bill segment FT algorithm of Payoff Amt = Bill Amt / Current Amt = Amt Due). Though not evident by the name, the Payoff Amt = Bill Amt / Current Amt = Amt Due algorithm does set the current amount to zero for monitored Non-Billed Budgets. (For SAs with other roles, the current amount is equal to the amount due or the recurring charge.)

A bill completion algorithm transfers money from the Non-Billed Budget to the covered SAs (if you plug in the NBB Credit Transfer bill completion algorithm on the Non-Billed Budget SA type). The algorithm type supplied with the base package distributes the credit using the method described in Distributing Non-Billed Budget Credit.

NOTE:
Canceled Bill Segments. No new processing occurs when a bill segment is canceled; any credit balance remains on the covered SA.

Distributing Non-Billed Budget Credit

Both the NBB Credit Transfer bill completion algorithm type and the Non-Billed Budget SA stop algorithm type supplied with the base package use the same method of distributing a credit from a Non-Billed Budget SA to the covered SAs. The following points describe how the credit is distributed:

- Covered SAs that are already in credit (due to some other circumstance, such as a cancellation and rebill) are excluded from the distribution.
- The distribution to each covered SA will not exceed its total payoff to ensure that none of the covered SAs have a credit balance.

- The credit is prorated over the covered SAs according to the relative payoff balances on each SA.
- The calculation of the payoff balance is adjusted to exclude the current balance to ensure that the credit is prorated over the debt covered by the budget, not any ad-hoc debt for the SA.

> **NOTE:**
> The current balance on covered SAs should always be zero. The only exception occurs if an adjustment has been added that directly affected the SA balance. In this case, the distribution assumes that the balance is outside the non-billed budget and needs to be paid separately.

- Any excess credit remains on the Non-Billed Budget SA until the next distribution takes place or until the Non-Billed Budget SA is stopped.
- The type of adjustments created is determined by the **Adjustment Type (Xfer)** specified on the Non-Billed Budget SA type.

The examples in the table below illustrate the points above.

NBB SA Credit	SA 1 Payoff	SA 1 Current	SA 2 Payoff	SA 2 Current	SA 1 Credit Xfer	SA 2 Credit Xfer
-100.00	-100.00	0.00	-200.00	0.00	0.00	0.00
-100.00	150.00	0.00	-50.00	0.00	-100.00	0.00
-300.00	150.00	0.00	50.00	0.00	-150.00	-50.00
-100.00	150.00	0.00	250.00	0.00	-37.50	-62.50
-100.00	150.00	0.00	250.00	50.00	-42.86	-57.14

The first example above shows that covered SAs with a credit balance are excluded from the distribution. Any excess credit remains on the Non-Billed Budget.

The second example shows that one covered SA has a credit balance, so the entire credit is distributed to the remaining SA.

The third example shows the amount distributed to a covered SA does not exceed its payoff balance. Again, any excess credit remains on the Non-Billed Budget SA.

The fourth example illustrates how the credit is prorated based on the payoff balance. The fifth example illustrates the same prorating but with a current balance on one of the SAs (SA 2). (Remember that the prorating excludes any current balance.)

The prorated amount is calculated by subtracting the current balance from the payoff balance then multiplying the result by the distribution amount and dividing by the total payoff owing of all covered SAs.

Stopping an SA Covered By a Non-Billed Budget

If a service agreement covered by a Non-Billed Budget is stopped, the system must bring the current balance and payoff balance of the covered SA back in synch. The system creates an adjustment using the Synch Current adjustment type from the SA's SA type.

In the example below, the electric SA's payoff balance is 21. The first twelve transactions are discussed above.

Event	Effect On Current Balance	Effect On Payoff Balance	Current Balance	Payoff Balance	Effect On Current Balance	Effect On Payoff Balance	Current Balance	Payoff Balance
	Electric SA				Non-Billed Budget SA			

Starting balance	0	0	25	25				
Non-Billed Budget is activated	-25	0	0	25				
First scheduled payment is due					10	0	10	0
Payment					-10	-10	0	-10
Second scheduled payment is due					10	0	10	-10
Over-payment					-10	-10	0	-20
Third scheduled payment is due					10	0	10	-20
Transfer Adjustment					-10	-10	0	-30
Fourth scheduled payment					10	0	10	-30
Underpayment					-7	-7	3	-37
Bill	0	33	0	58				
Bill completion (transfer adjustment)	0	-37	0	21	0	37	3	0
SA is stopped	21	0	21	21				

NOTE:
In addition to synching the current and payoff balance, the SA being stopped is removed from the covered SAs collection. When the last covered SA is removed from the collection, the Non-Billed Budget is also stopped.

Financial Transactions For Unmonitored Non-Billed Budgets

Some companies have the concept of non-billed budgets where the payments made by the customer are optional. This functionality is implemented as an unmonitored Non-Billed Budget. Unmonitored Non-Billed Budgets allow a customer to make optional prepayments towards a bill.

As explained previously, unmonitored Non-Billed Budgets receive different financial treatment than monitored Non-Billed Budgets. The financial transaction algorithms use the **Non-Billed Budget Monitor** flag on the Non-Billed Budget's SA type to create the appropriate financial transactions. The following table describes the differences in the financial treatment of monitored and unmonitored Non-Billed Budgets.

System Event	Monitored Non-Billed Budgets	Unmonitored Non-Billed Budgets
When a Non-Billed Budget is activated or an SA is added to a Non-Billed Budget...	The system creates adjustments for all affected SAs to set the current balance equal to zero.	The current balances for covered SAs are not zeroed. The activate SA and Non-Billed Budget maintenance processing do not create any adjustments if the Non-Billed Budget is unmonitored.
When a scheduled payment is due...	An adjustment is created to increase the Non-Billed Budget's current balance by the expected amount.	There is no change to the Non-Billed Budget's current balance; it is always zero. The scheduled payment processing background process does not execute the NBB process scheduled payment algorithm.
When the payment is made...	The Non-Billed Budget's current balance is reduced to zero and the Non-Billed Budget's payoff balance reflects the accumulated credit from the payment.	Typically the payments are distributed to an excess credit (overpayment) SA. Refer to Distributing Payments for Unmonitored Non-Billed Budgets for more information.
When a bill segment financial transaction is created...	The current amount is set to zero for SAs that are covered by the Non-Billed Budget.	The covered SAs' current amounts are equal to their payoff amounts.
At bill completion...	Money from the Non-Billed Budget is transferred to the covered SAs. Customers may receive a bill for information purposes, but they are not required to pay it.	Money from the overpayment SA is transferred to the account's SAs. The customer is still liable to pay the outstanding balance.
When the Non-Billed Budget is stopped or an SA is removed from a Non-Billed Budget...	The system creates adjustments for all affected SAs to synchronize their current balances with their payoff balances, thus removing the zero current balance and replacing it with the actual current balance.	There is no change to the current balances for covered SAs as the balances already reflect the actual current balance.

Unmonitored Non-Billed Budgets also receive different credit and collections treatment. Refer to Credit and Collections and Unmonitored Non-billed Budgets for more information.

Distributing Payments for Unmonitored Non-Billed Budgets

For Non-Billed Budgets, payments are distributed according to the payment and overpayment algorithms on the customer class. The base package payment distribution algorithm applies a payment first towards any SAs that have overdue or current balances (refer to Distributing A Payment for more information). Since the unmonitored Non-Billed Budget SA doesn't have a current balance, it is not considered by the payment distribution algorithm. If there aren't any SAs with a current balance, the overpayment distribution algorithm handles the remaining credit. You can elect to:

- Apply the overpayment to an excess credit SA. This is the method that we strongly recommend because all financial transactions are then a function of the normal payment, overpayment and billing processes.

- Apply the overpayment to the highest priority SA that is eligible for overpayment (as specified on the SA type). You can use this method to apply the overpayment to the unmonitored Non-Billed Budget SA. If you use this method, you must also set up the system to transfer the credit from the unmonitored Non-Billed Budget.

NOTE:
Use An Excess Credit SA. We strongly recommend that payments for unmonitored Non-Billed Budgets are distributed to an excess credit SA. In this case, the Non-Billed Budget is just a shell to hold the covered SAs and recommend a payment schedule; all financial transactions are a function of the normal payment, overpayment and billing processes.

FASTPATH:
Refer to Overpayment Segmentation for a detailed discussion of overpayment distribution options.

Transferring Credit from Unmonitored Non-Billed Budgets

If you distribute an overpayment to an unmonitored Non-Billed Budget SA (i.e. the unmonitored non-billed budget maintains a credit balance instead of an overpayment SA), you must plug-in a bill completion algorithm on the SA type to transfer the credit balance to the covered SAs at bill completion time. The bill completion algorithm type (BCMP-NB) supplied with the base package transfers the credit balance to the covered service agreements when the bill is completed. Additionally, the **Adjustment Type (Xfer)** on unmonitored Non-Billed Budget SA types should reference a FT algorithm of Payoff Amt = Adj / Current Amt = Adj to ensure that the credit is removed from both the current and payoff balances.

WARNING:
You must create your own SA stop algorithm type for correctly stopping an unmonitored non-billed budget that maintains a credit balance. The SA stop algorithm that is supplied with the base package does NOT transfer remaining credit from the unmonitored Non-Billed Budget SA. (The base package SA stop algorithm transfers the remaining credit using the overpayment distribution algorithm on the customer type, which you have set up to transfer to back to the unmonitored non-billed budget.)

Designing Non-Billed Budgets

The topics in this section describe functionality that you must consider when designing non-billed budgets.

Making SAs Eligible For Non-Billed Budgets

A billable SA may be covered by a Non-Billed Budget when its SA type is flagged as Eligible for Non-Billed Budget.

All SAs that are eligible for non-billed budgets should reference a bill segment type that uses the Payoff Amt = Bill Amt / Current Amt = Amt Due bill segment FT algorithm. This algorithm sets the SA's current amount to zero if it is covered by a monitored Non-Billed Budget.

A list of the SAs covered by a Non-Billed Budget is maintained with the non-billed budget SA. This list is used at bill completion to determine the financial transactions that should occur.

Designing Recommendation Rules

Users (and the renewal process) ask the system to recommend the scheduled payments for a Non-Billed Budget. In general, this recommendation process must establish:

- The amount to be paid
- The dates on which the payments are due

We envision many different types of recommendation rules. For example:

- Recommend 26 scheduled payments to be made on a fortnightly basis that are due on Tuesdays.
- Recommend monthly payments that are due on the nearest workday after the 10th of the month.
- Recommend scheduled payments where the customer pays their annual charges in 10 out of 12 months where the payments are not due in November and December.
- Recommend bi-monthly payments where the payments are due on the first workday following the 5th and 20th of the month.

Additionally, the recommendation rules must determine how to handle any outstanding payoff balances for covered SAs. The true-up rule provided with the base package payment schedule algorithm type can ignore the payoff balance, divide the payoff balance evenly between the scheduled payments, or add the payoff balance to the first scheduled payment.

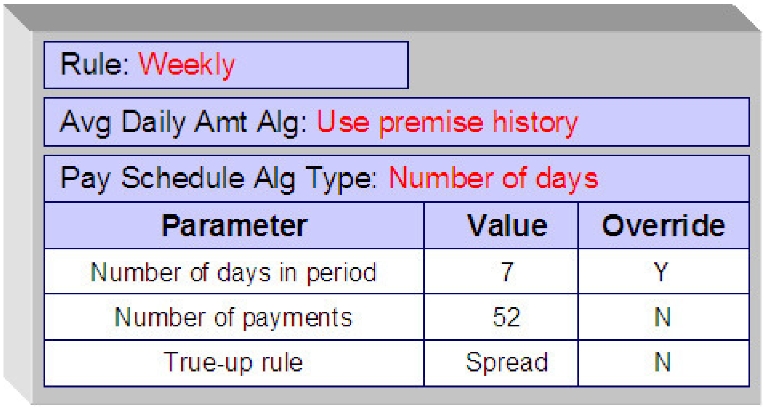
A recommendation rule comprises three elements:

- An algorithm to calculate an average daily amount (the NBDA-DA algorithm type provided with the base package uses premise billing history)

> **NOTE:**
> **Calculating an Amount for Non-utility SAs.** The algorithm type supplied with the base package only handles service-point oriented SAs. For example, the average daily amount algorithm calculates an average daily amount. For non-utility SAs, you must develop the appropriate algorithm types.

- Two algorithm types are provided with the base package to calculate a schedule of payments. NBPS-MON calculates a monthly schedule and NBPS-PS calculates a scheduled based on a specified number of days.
- A collection of default parameter values for the payment schedule algorithm type

A recommendation rule (based on the algorithm types provided with the base package) is illustrated below.



Rule: Weekly

Avg Daily Amt Alg: Use premise history

Pay Schedule Alg Type: Number of days

Parameter	Value	Override
Number of days in period	7	Y
Number of payments	52	N
True-up rule	Spread	N

Weekly Payments Recommendation Rule

➤ **NOTE:**

Additional Parameters. Not all of the parameters associated with the weekly payment schedule algorithm type are illustrated. Refer to the NBPS-PS algorithm type for a detail description of the parameters.

The default parameter values for the payment schedule algorithm type may change over time, so the collection contains an effective date. If default values are changed, these changes do not affect Non-Billed Budgets already in effect. Existing Non-Billed Budgets keep the parameter values that were used when the Non-Billed Budget was started.

A user may override the default parameter values for the payment schedule algorithm type to customize the schedule if an override is allowed for a parameter. Additionally, a user may edit the payment schedule details at any time (provided the payment has not yet been processed).

The parameter values used for the recommendation rule are kept with the Non-Billed Budget SA, so that any customized parameter changes can be re-applied to a renewed Non-Billed Budget. For example, the parameter that determines the payment due day may default to the first of the month. To customize the schedule, this value may be changed to the fifth of the month. This amended value is kept with the Non-Billed Budget SA to ensure that the renewed budget follows the same monthly schedule.

➤ **NOTE:**

Normally parameter values for an algorithm type are kept with the algorithm. Because the parameters may vary for each Non-Billed Budget, the parameter values are kept with the SA in the case of Non-Billed Budgets.

Refer to Non-billed Budgets Recommendation Rule - Main for information on creating recommendation rules.

Example Recommendation Rules

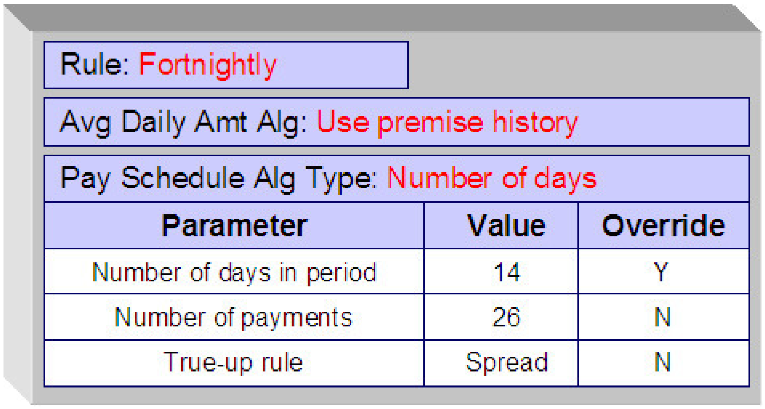
The following examples may be helpful in designing and implementing your recommendation rules.

➤ **NOTE:**

Developing Your Own Payment Schedule Algorithms. The base package comes supplied with a monthly payment schedule algorithm type. You can use this algorithm as an example when creating payment schedule algorithm types for your implementation.

Fortnightly Payments Recommendation Rule Example

The following diagram illustrates a recommendation rule where the customers pay every two weeks. The current balance for any covered SAs is added to the first payment.



Rule: Fortnightly

Avg Daily Amt Alg: Use premise history

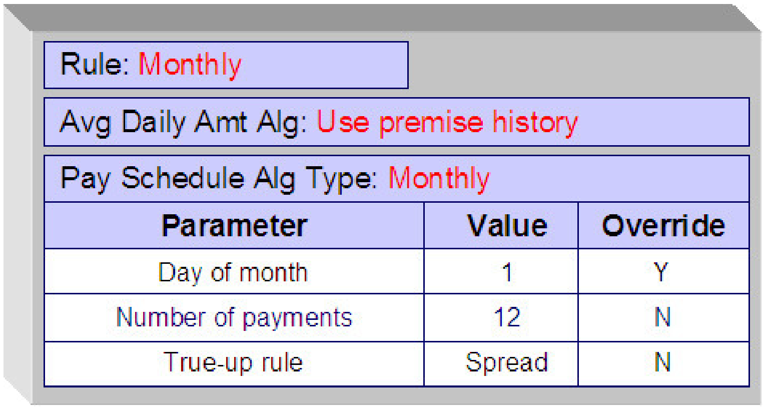
Pay Schedule Alg Type: Number of days

Parameter	Value	Override
Number of days in period	14	Y
Number of payments	26	N
True-up rule	Spread	N

Fortnightly Payments Recommendation Rule

Monthly Payments Recommendation Rule Example

The following diagram illustrates a recommendation rule where the customers pay twice monthly on the first of the month. The current balance for any covered SAs is spread out over the scheduled payments.



Rule: Monthly

Avg Daily Amt Alg: Use premise history

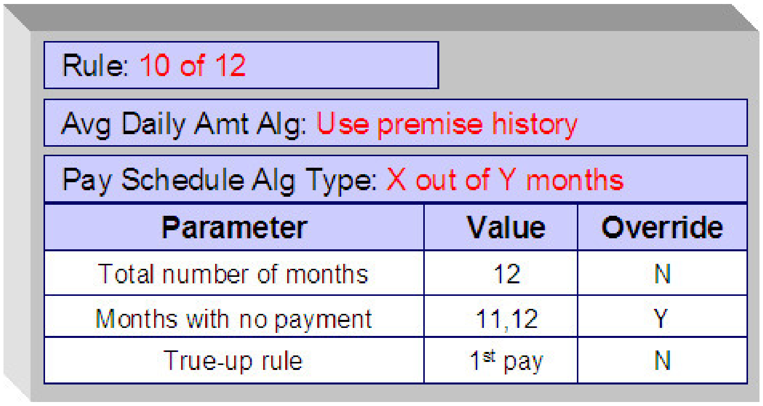
Pay Schedule Alg Type: Monthly

Parameter	Value	Override
Day of month	1	Y
Number of payments	12	N
True-up rule	Spread	N

Monthly Payments Recommendation Rule

Ten Out of Twelve Months Recommendation Rule Example

The following diagram illustrates a recommendation rule where the customers pay one a month except for months during a holiday season (November and December). The current balance for any covered SAs is added to the first payment.



Rule: 10 of 12

Avg Daily Amt Alg: Use premise history

Pay Schedule Alg Type: X out of Y months

Parameter	Value	Override
Total number of months	12	N
Months with no payment	11,12	Y
True-up rule	1st pay	N

Ten Out of Twelve Months Recommendation Rule

Activating Non-Billed Budgets

You can plug in an algorithm (of type SACR-AT) on the SA Creation system event on the Non-Billed Budget SA type to automatically activate the Non-Billed Budget (i.e. transition it from pending start to active status). If you don't use a SA Creation algorithm to activate the Non-Billed Budget, it is activated the next time the SA activation background process runs.

When a Non-Billed Budget is activated, you can perform special processing using an algorithm plugged in on the SA Activation system event on the Non-Billed Budget SA type. The special processing can be developed to do anything that you would like, for example you could:

- Create a customer contact that with an appropriate letter template can generate a letter to inform the customer of their payment amount and payment schedule.
- Initiate the creation of a payment coupon book for a customer.

The system comes supplied with a sample algorithm type (called SAAT-CC) that simply creates a customer contact to indicate that the Non-Billed Budget is activated.

Renewing Non-Billed Budgets

A Non-Billed Budget can be renewed either manually or via a background process. When the Non-Billed Budget SA is created, the expiration date, renewal date and the recommendation rule used to create the initial budget are kept with the SA. A renewal flag on the Non-Billed Budget SA type controls if a renewal is required, optional or not allowed. If renewal is required, a user must specify a renewal date when creating the service agreement. The renewal date is defaulted on to an SA based on the valued of the **Days Before Expiration for Renewal** field on the SA type.

An algorithm on the SA type can customize the processing required to renew an SA.

The SA renewal background process:

- Executes the SA renewal algorithm (specified on the SA type) when the renewal date is reached (i.e., it is on or before the process date). The base package comes with an algorithm type (SARN-NB) that determines the current recommendation rule for a non-billed budget and executes the associated payment schedule algorithm using the Non-Billed Budget SA-specific parameter values to generate a new schedule. It returns new expiration and renewal dates.
- If the renewal algorithm is successful, the renewal and expiration date fields on the SA are updated with the new values.
- If the renewal process is not successful, a To Do list entry (of type TD-SARN) is created for the account and SA.

The new payment schedule that is returned from the renewal process for a Non-Billed Budget is appended to the current schedule.

A user can manually launch the renewal process for a Non-Billed Budget SA by clicking the **Renew NBB** button on the Non-Billed Budget maintenance page.

Expiring Non-Billed Budgets

Non-Billed Budget service agreements may specify an expiration date. The SA expiration background process initiates the stop process for all pending start or active SAs where the expiration date is reached (before or on the process date).

Stopping Non-Billed Budgets

When a Non-Billed Budget stop is initiated, either on request or because it has expired and is not being renewed, the Non-Billed Budget is transitioned to pending stop status. You can plug in an algorithm (of type SAIS-ST) on the SA Stop Initiation system event on the Non-Billed Budget SA type to automatically finalize and stop the SA (i.e. transition it to stopped status). If you don't use a SA Stop Initiation algorithm, the Non-Billed Budget is stopped the next time the SA activation background process runs.

To finalize a pending stop SA, the system first calls the stop SA algorithm plugged-in on the SA Stop system event on the SA type. The stop SA algorithm type (SAST-NB) supplied with the base package:

- Distributes any credit on the Non-Billed Budget to the covered SAs (using the method described in Distributing Non-Billed Budget Credit)
- Distributes any excess credit remaining on the non-billed budget using the overpayment distribution algorithm for the account's customer class and the overpayment transfer adjustment type (specified as a parameter to the algorithm)
- Creates a trigger to cause the account to be reviewed by the account debt monitor
- Creates a customer contact (if the customer contact class and customer contact type parameters are populated)

WARNING:
If you do not plug in an SA stop algorithm that transfers the credit balance from the non-billed budget to its covered SAs (or an excess credit SA), the stopped Non-Billed Budget may have a credit balance. You must then manually distribute this credit.

After the SA stop algorithm is finished, the SA stop processing performs the following steps if the SA type has a special role of Non-Billed Budget:

- If the Non-Billed Budget is monitored, create adjustments to synchronize the current and payoff balance of covered SAs using the **Adj. Type (Synch Current)** adjustment type from the covered SAs' SA types
- Remove the covered SAs from the Non-Billed Budget
- Remove all the scheduled payments from the non-billed budget
- Create an adjustment to synchronize current and payoff on the Non-Billed Budget SA using the **Adj. Type (Synch Current)** adjustment type from the Non-Billed Budget's SA type

➤ **NOTE:**

Synchronizing current and payoff effectively sets the current amount to zero on the Non-Billed Budget SA, as the payoff amount should have been reduced to zero by the distribution and overpayment processing in the algorithm for SA Stop.

●

FASTPATH:
Refer to The Lifecycle Of A Service Agreement for more information about how a pending stop SA is stopped and closed.

Automatic Payment and Non-Billed Budgets

If a customer wants to pay their non-billed budget scheduled payments automatically, the account must be set up for automatic payment. In addition, the Non-Billed Budget must indicate that automatic payment is being used.

●

FASTPATH:
Refer to How To Set Up Automatic Payment For A Non-Billed Budget for more information.

When this is done, a background process referred to as NBBAPAY creates automatic payments on the scheduled payment date by calling the automatic payment creation algorithm plugged in on the installation record.

➤

NOTE:
You must ensure that your auto pay creation algorithm supports Non-Billed Budget scheduled payments. The APAY-CREATE algorithm type supplied with the base package supports non-billed budget scheduled payments.

Credit and Collections and Non-Billed Budgets

Unless the Non-Billed Budget is unmonitored, the account debt monitor (ADM) monitors a Non-Billed Budget SA's current amount just as it does for any other SA. The scheduled payment algorithm creates a trigger to ensure that the account debt monitor reviews the account the next time it runs. The review date on the trigger record is set to the process business date.

A separate debt class is needed for Non-Billed Budget SA types, thus allowing you to define collection class controls, debt criteria and collection process templates specifically for Non-Billed Budgets. The debt criteria should be set up to trigger a collection process when the arrears amount exceeds $0.01 for more than n payment periods plus the number of grace days that you want to allow.

The collection process template can perform any of the events in standard collection processes, such as sending letters to customers and creating severance processes. At a minimum, the collection process template should be set up to start a severance process for all service agreements in the debt class. (Since the debt class is specifically for Non-Billed Budgets, the Non-Billed Budget is the only SA that will be subject to a severance process.)

The Non-Billed Budget severance process template should include the following event types:

- Populate a characteristic on the SA to indicate that the SA is broken. The base package comes supplied with a severance event algorithm type (SVEV-NB) that sets an SA characteristic to indicate that it was "severed".

- Expire Severance Agreement to move the SA to the pending stop state.

When the system subsequently stops the Non-Billed Budget, the system removes the covered SAs from the Non-Billed Budget and synchronizes their current balances with their payoff balances. Since the SAs have current balances again, they are subject to the account debt monitor, which can start subsequent collection processes for any of the SAs that meet the debt criteria for their debt class.

●

FASTPATH:
Refer to Stopping Non-billed Budgets for a complete description of the events that occur when a Non-Billed Budget is stopped.

Customers can catch up on their payments and avoid having their Non-Billed Budget broken as long as their current balance doesn't violate the debt criteria for the Non-Billed Budget's collection process.

Credit and Collections and Unmonitored Non-Billed Budgets

If the **Non-Billed Budget Monitor** flag on the Non-Billed Budget's SA type is set to Unmonitored, the NBB Scheduled Payment Processing background process does not create a trigger for the account debt monitor. Additionally, the Non-Billed Budget's current amount is always equal to zero, so it never violates any debt collection criteria. We recommend using a debt class that has the **Eligible for Collection** flag turned off, such as the N/A debt class.

For unmonitored Non-Billed Budgets, the current balance is kept on the covered SAs so they are subject to the account debt monitor and any debt criteria for their SA types' debt classes. For more information, refer to Financial Transactions for Unmonitored Non-Billed Budgets.

Non-Billed Budget Status

A Non-Billed Budget SA's status is just like any other SA's status. In addition, you can use a characteristic to keep an explicit status relevant to the Non-Billed Budget:

- A base package severance event algorithm type (SVEV-NB) creates a characteristic value to indicate if a Non-Billed Budget is "broken" (i.e. stopped as a result of a severance process).
- A base package break Non-Billed Budget algorithm type (NBBR-BRK) creates a characteristic value to indicate if a Non-Billed Budget is "canceled" (i.e. manually stopped by a user).

An active Non-Billed Budget implicitly has a "kept" status (i.e. all scheduled payments have been made).

Alerts For Non-Billed Budgets

The system provides alerts to highlight the existence of Non-Billed Budgets. These alerts are important to assist the customer service representatives:

- An alert is displayed if the account has a Non-Billed Budget that is not stopped (e.g., pending start, active or pending stop).
- When a user denies a Non-Billed Budget (for whatever reason), the user should create a customer contact with a given customer contact type. This type of alert prevents the customer from shopping around. An existing alert algorithm type (CC BY TYPCL) can highlight these customer contacts.

- For customers who are permanently forbidden from having a Non-Billed Budget, the user should put a permanent alert on the account.
- Use an algorithm to highlight cancelled or severed non-billed budgets with an entry in the alert zone. The algorithm type to do this is not provided. Use PP BY STATUS and CCAL-WF as examples of how to create this type of algorithm.

> ●
> **FASTPATH:**
> For more information about introducing alert conditions on Control Central, refer to Installation Options - Algorithms.

Non-Billed Budget Recommendation Rule

Recommendation rules are used to recommend scheduled payments for Non-Billed Budgets. For information about designing recommendation rules, refer to Designing Recommendation Rules.

To define recommendation rules, navigate to **Admin** > **Financial** > **NBB Recommendation Rule** > **Add.**

Description of Page

Enter an easily recognizable **Recommendation Rule** code and **Description** for each recommendation rule.

Specify the **Average Daily Amount Algorithm** used to calculate the average daily amount for this recommendation rule.

Specify the **Payment Schedule Algorithm Type** used to create the recommended payment scheduled for Non-Billed Budgets that use this recommendation rule. The Payment Schedule Algorithm Type cannot be modified if a Non-Billed Budget SA that is not stopped or cancelled is using this recommendation rule.

Payment Schedule Parameters enables you to define collections of default parameter values for the payment schedule algorithm type that are effective dated. For each collection:

- **Effective Date** defines the date on which the collection of parameter values becomes effective.
- **NBB Rule ParameterValue** specifies the default value of each parameter supplied to the algorithm. Note that the payment schedule algorithm type controls the number and type of parameters.
- **Override Flag** indicates whether the user can override the default value for the parameter.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference CI_NB_RULE.

Setting Up The System To Enable Non-Billed Budgets

The above topics provided background information about how Non-Billed Budgets are supported in the system. The topics in this section describe how to set up the system to enable Non-Billed Budget functionality.

> ➤
> **NOTE:**
> **Example Setup.** This section describes typical non-billed budget configurations. Your set up and configuration may differ depending on your business needs. This section is provided for guidance only. Read

> the descriptions of Non-Billed Budget functionality above to understand the implications of the described setup.

NBB Distribution Codes

You must set up a Non-Billed Budget Clearance distribution code that is used on the Non-Billed Budget SA type to credit non-billed payments.

NBB Adjustment Types

> **NOTE:**
> **Non-Billed Budget Financial Transaction Algorithms.** The Non-Billed Budget adjustment types use the standard FT algorithm types that are provided with the base package. If you have not yet defined algorithms for these types in your system, do so before creating the Non-Billed Budget adjustment types.

The following adjustment types are needed:

- Add SA To Non-Billed Budget. This adjustment type should reference an FT algorithm of Payoff Amt = 0 / Current Amt = Adj Amount (no GL). Because the adjustment affects the current balance only, there is no entry in the GL. This adjustment type is referenced on **Adjustment Type (Current=0)** on SA types that are eligible for Non-Billed Budgets. Note that this adjustment type is never used if the SA is added to an unmonitored Non-Billed Budget.
- Bill Complete for Monitored Non-Billed Budget. This adjustment type should reference a distribution code used for balance transfer clearing. It should reference a FT algorithm of Payoff Amt = Adj / Current Amt = 0. This adjustment type is referenced on **Adjustment Type (Xfer)** on SA types that are monitored Non-Billed Budgets.
- Synch Balance for Non-Billed Budget. This adjustment type should be set up for **Sync. Current Amount** and should reference a FT algorithm of Payoff Amt = 0 / Current Amt = Adj Amount (no GL). This adjustment type is referenced on the **Adj. Type (Synch Current)** on Non-Billed Budget SA types. It is used when a Non-Billed Budget SA is stopped to synch the current balance with the payoff balance.
- Scheduled Payment. This adjustment type should reference an FT algorithm of Payoff Amt = 0 / Current Amt = Adj Amount (no GL). Because the adjustment affects the current balance only, there is no entry in the GL. This adjustment type is referenced on the Non-Billed Budget process scheduled payment algorithm.
- Overpayment Transfer. This adjustment type should reference a transfer distribution code and a FT algorithm of Payoff Amt = Adj / Current Amt = Adj. This adjustment is referenced on the non-billed budget process scheduled payment algorithm and the stop non-billed budget algorithm. Note that if there is already a transfer adjustment type created in your system, you do not need to create a new one.

If you are setting up an unmonitored Non-Billed Budget that maintains a credit balance (as opposed to maintaining the credit balance on an overpayment SA), you need to create an adjustment type for Bill Complete for Unmonitored Non-Billed Budget. (Refer to Transferring Credit from Unmonitored Non-billed Budgets for more information.) The adjustment type should reference a transfer distribution code and a FT algorithm of Payoff Amt = Adj / Current Amt = Adj. This adjustment type is referenced on **Adjustment Type (Xfer)** on SA types that are unmonitored Non-Billed Budgets.

NBB Adjustment Type Profiles

Create an adjustment type profile for Non-Billed Budgets that references the following adjustment types:

- Bill complete for monitored Non-Billed Budget
- Bill complete for unmonitored Non-Billed Budget (if used)
- Synch balance for Non-Billed Budget
- Scheduled payment
- Overpayment transfer

Create an adjustment type profile for eligible SAs that references the following adjustment types:

- Add SAs to monitored Non-Billed Budgets
- Bill complete for monitored Non-Billed Budget
- Bill complete for unmonitored Non-Billed Budget (if used)

➤ **NOTE:**

Bill Complete Adjustment Types. Because the bill complete adjustment types transfer amounts between two SAs, they must be in profiles for both Non-Billed Budget and eligible SA types.

Overpayment Transfer Adjustment Type. The overpayment transfer adjustment type created above is used to transfer funds from an excess credit SA to a Non-Billed Budget when the scheduled payment is processed. It is also used to transfer excess funds from a Non-Billed Budget that is being closed to an excess credit SA. The transfer adjustment should therefore be added to an adjustment type profile that is referenced on the excess credit SA type.

NBB Characteristic Types

Create a Non-Billed Budget status characteristic type that specifies Service Agreement as its characteristic entity. The characteristic type should include the following predefined values:

- Non-Billed Budget Canceled
- Non-Billed Budget Severed

NBB Customer Contact Class And Types

Create a Non-Billed Budget customer contact class. The contact class may include the following customer contact types:

- Non-Billed Budget Activate
- Non-Billed Budget Renewal
- Non-Billed Budget Stop

➤ **NOTE:**

Customer Contact Letters. If you want to send letters to your customers when a contact of any of these types is created, you must create an appropriate letter template and attach it to the contact type.

NBB Algorithms

You must define the following algorithms:

- On NBB recommendation rule, the Non-Billed Budget Daily Amount Calculation. Refer to the NBDA-DA algorithm type for more information about the base package algorithm.
- On SA type:
  - Non-Billed Budget Process Scheduled Payment. This algorithm has parameters that must reference the Scheduled Payment and Overpayment Transfer adjustment types defined above. Another parameter references an overpayment SA type (that you may need to create if there is not already one in your system). Refer to the NBPA-PS algorithm type for more information about the base package algorithm.
  - Non-Billed Budget SA Renewal. Refer to the SARN-NB algorithm type for more information about the base package algorithm.
  - Break Non-Billed Budget SA. Refer to the NBBR-BRK algorithm type for more information about the base package algorithm.
  - SA Activation - automatically activate SA (if you want to automatically activated Non-Billed Budgets when they are created). Refer to the SACR-AT algorithm type for more information about the base package algorithm. Note that this same algorithm may be used on many SA types.
  - SA Activation - create customer contact (if you want the system to create a customer contact when NBB SAs are activated). Refer to the SAAT-CC algorithm type for more information about the base package algorithm.
  - SA Stop - automatically stop SA (if you want to automatically transition Non-Billed Budgets from pending stop to stop when their stop is initiated). Refer to the SAIS-ST algorithm type for more information about the base package algorithm. Note that this same algorithm may be used on many SA types.
  - SA Stop - Stop Non-Billed Budget. Refer to the SAST-NB algorithm type for more information about the base package algorithm.
  - Bill Completion - Non-Billed Budget Credit Transfer. Refer to the BCMP-NB algorithm type for more information about the base package algorithm.
- On bill segment type, the Bill FT Algorithm (for all SAs that are eligible for NBB). Refer to the BSBF-BA algorithm type for more information about the base package algorithm. Note this is the standard bill FT algorithm type used for common bill transactions. It supports bill FT for non-billed budgets and other SA types. You only need to create it if it doesn't already exist in your system.
- On severance event type, an algorithm for Non-billed Budget Severance. Refer to the SVEV-NB algorithm type for more information about the base package algorithm.

➤ **NOTE:**
Payment Schedule Algorithm Types. For Non-Billed Budget payment schedule algorithm types, you need to define the algorithm types (if you add your own algorithm types). You do not need to define algorithms because the parameter values for the algorithm are defaulted on the recommendation rule and stored with the Non-Billed Budget SA. (Normally the algorithm holds the parameter values.) Refer to the NBPS-MON algorithm type for more information about the base package payment schedule algorithm.

NBB Debt Class And Collection Process

Set up a separate debt class, collection class control, collection process template and severance process template for Non-Billed Budgets according to the information in Credit and Collections and Non-Billed Budgets.

SA Types for SAs Covered by NBBs

You must modify the SA type for any SAs that you want to allow to be covered by a Non-Billed Budget.

- Verify that the specified **Bill Segment Type** (on the Billing page) references a financial transaction algorithm to set the current amount to zero for monitored Non-Billed Budgets. The FT creation algorithm type BSBF-BA (i.e., Payoff Amt = Bill Amt / Current Amt = Amt Due) supplied with the base package sets the current amount to zero for SAs that are covered by monitored non-billed budgets, though this is not evident by the name. (For SAs that are not covered by a Non-Billed Budget, the current amount is equal to the amount due or the recurring charge.) Refer to Billing For SAs Covered By The Non-Billed Budget for more information.
- Set the **Eligible for Non Billed Budget** flag (on the Billing page) to Eligible for Non-Billed Budget .
- Populate the **Adjustment Type (Current = 0)** (on the Detail page) to indicate the adjustment to be used to zero out the current amount on the covered SAs when the Non-Billed Budget SA is activated. Use the Add SA To Non-Billed Budget adjustment created above. This adjustment type is only called for SAs that are covered by a monitored Non-Billed Budget.
- Reference an **Adjustment Type Profile** (on the Adjustment Profile page) that includes the Add SA To Non-Billed Budget adjustment type referenced in the **Adjustment Type (Current = 0)** field above.
- If not already specified (for write off), an **Adj. Type (Synch Current)** (on the Main page) is also required. It is used to synchronize (make equal) the current amount with the payoff amount when the SA is removed from (i.e. no longer covered by) a Non-Billed Budget.

NBB Recommendation Rules

Set up any NBB Recommendation Rules that you want to be available for your Non-Billed Budget SAs. Use any Non-Billed Budget Daily Amount Calculation algorithms defined above. Also use the Non-billed Budget Monthly Payment Schedule (NBPS-MON) algorithm types and specify the default parameters.

Non-Billed Budget SA Types

You must set up a SA types for your Non-Billed Budget service agreements. You may need multiple Non-Billed Budget SA types if you have different business rules for different types of Non-Billed Budgets, for example, if you have both monitored and unmonitored Non-Billed Budgets or if you support Non-Billed Budgets with different renewal requirements. The following points provide guidelines for creating a non-billed budget SA type.

SA Type - Main (NBB)

Service Type should reference something like "Miscellaneous Service".

Distribution Code should be the one you set up to book credits for Non-Billed Budgets.

Revenue Class should be set to N/A. (Revenue classes are not applicable because Non-Billed Budgets do not apply a rate and revenue classes are only relevant for SA types that use a rate.)

The **Payment Segment Type** should reference the Normal Payment.

Do Not Overpay should be on. Any excess payments should go to the overpayment SA, not the Non-Billed Budget SA.

Late Payment Charge is not applicable and should not be turned on because Non-Billed Budgets are not billed.

Adj. Type (Synch Current) should reference the Synch Balance for Non-Billed Budget adjustment type created above.

SA Type - Detail (NBB)

- **Special Role** is Non-Billed Budget.
- **Adjustment Type (Xfer)** must be populated to indicate the adjustment to be used for transferring accumulated credit from the Non-Billed Budget SA to the covered SAs. This field is not used for unmonitored Non-Billed Budgets.
- **Renewal** may be optional, not allowed, or required depending on your business processes.
- If Renewal is required, specify the **Days Before Expiration for Renewal**.
- **Non-Billed Budget Monitoring** must indicate whether the Non-Billed Budget is monitored by the account debt monitor.

SA Type - Billing (NBB)

Non-Billed Budget SAs do not get billed, so the **Eligible for Billing** flag should be off.

Additionally, the **Characteristic Premise Required** should not be checked for Non-Billed Budgets.

SA Type - Rate (NBB)

Non-Billed Budget SAs do not use rates, so the **Rate Required** flag should be off.

SA Type - SP Type (NBB)

Non-Billed Budget SAs do not have service points, so the **Service Points Required** flag should be off.

SA Type - Adjustment Profile (NBB)

Adjustment Type Profile should reference the adjustment type profile for Non-Billed Budgets (set up above).

SA Type - Credit and Collections (NBB)

The credit and collections information should reference a **Severance Process Template** that calls the Non-Billed Budget Severance algorithm created above and expires the Non-Billed Budget SA. The **Debt Class** should reference an appropriate value consistent with your credit and collections rules. Typically, monitored Non-Billed Budgets should be in their own debt class. Unmonitored Non-Billed Budgets should have a **Debt Class** that is not Eligible for Collection.

The **Write Off Debt Class** is not applicable because the Non-Billed Budget contains no debt to be written off. Reference a debt class such as N/A. Refer to Defining Credit & Collections Options for more information.

SA Type - Algorithms (NBB)

The SA type algorithms defined above must be set up:

- The Non-Billed Budget Credit Transfer algorithm created above should be specified for the Bill Completion system event (not used on unmonitored Non-Billed Budgets).
- The Break Non-Billed Budget SA algorithm created above should be specified for the Break NBB SA system event.
- The Non-Billed Budget Process Scheduled Payment algorithm created above should be specified for the Process NBB Scheduled Payment system event (not used on unmonitored non-billed budgets).
- The Non-Billed Budget SA Renewal algorithm created above should be specified for the SA Renewal system event.
- If you created the Automatic SA Activation algorithm above, it should be specified for the SA Creation system event.
- The Non-Billed Budget SA Activation algorithm created above should be specified for the SA Activation system event.
- If you created the Automatic SA Stop algorithm above, it should be specified for the SA Stop Initiation system event.
- The Stop Non-Billed Budget algorithm created above should be specified for the SA Stop system event.

SA Type - NBB Recommendation Rule (NBB)

Add the recommendation rules defined above that are valid for SAs of this type. Also, indicate which recommendation rule should be used as the default.

NBB Background Processes

Ensure that the following background processes are scheduled:

- Non-Billed Budget Scheduled Payment Processing (NBBPS)
- Non-Billed Budget Scheduled Payment Automatic Payment Create (NBBAPAY)
- Service Agreement Renewal (SARENEW)
- Stop Expired Service Agreements (SAEXPIRE)

Defining Quotation Options

This section describes tables that must be set up before quotations can be created.

> **FASTPATH:**
> For more information, refer to The Big Picture of Quotations.

Setting Up SA Types For Quotes

The topics in this section describe additional setup responsibilities required on SA types that can have proposal service agreements.

> **NOTE:**
> **Assumption.** We have assumed that you've already designed your SA types for the services that you sell. If you haven't done this, refer to Defining Service Agreement Types.

Enabling The Automatic Generation Of Billing Scenarios

As described under Proposal SAs Contain Billing Scenarios And Template Consumption, a proposal SA requires billing scenarios before a quote detail can be generated. The system can automatically create billing scenarios when a proposal SA is created. If this is required for your implementation, create a Proposal SA Creation algorithm type, and plug in the appropriate algorithm on each applicable Proposal SA type. Refer to PSAC-CBS for an example algorithm ***(note, this algorithm is not applicable for C2M but may be used as a reference only)***.

Enabling The Generation Of Simulated Bill Segments

As described under Creating Quotes And Quote Details, in order for the system to generate simulated bill segments for a proposal SA, you must plug-in a Proposal SA Bill Segment Generation algorithm on the proposal SA's SA type. Refer to CBSP-AR for an example algorithm that creates a simulated bill segments for each billing scenario linked to the proposal SA by calling rate application.

Enabling The Creation Of A Real SA When A Quote Detail Is Accepted

As described under Accepting / Declining Quote Details, in order for the system to create a real SA when a proposal SA is accepted, you must plug-in a Proposal SA Acceptance algorithm on the proposal SA's SA type. Refer to PSAA-PS for an example algorithm that creates a real SA by copying the proposal SA.

➤

NOTE:
Interval pricing service agreements. are not supported. PSAA-PS has a parameter for creating interval profiles, time-of-use maps and contract options when the real SA is created. This specific parameter/option is not applicable for C2M.

Setting Up Quote Route Types

Quote route types control how quotes are routed to customers and prospects. To define a quote route type, open **Admin** > **Sales & Marketing** > **Quote Route Type.**

●

FASTPATH:
Refer to Printing Quotes for more information about how quotes are routed to customers and prospects.

Description of Page

Enter a unique **Quote Route Type** and **Description** for every quote route type.

Quote Routing Method controls the type of information that may be defined when the respective **Quote Route Type** is selected on Account - Person Information. The following options are available:

- Postal: Use this method if the routing is via the postal service.
- Fax: Use this method if the routing is via fax.
- Email: Use this method if the routing is via email.

➤

NOTE:
The values for **Quote Routing Method** are customizable using the Lookup table. This field name is QTE_RTG_METH_FLG .

- The next two fields control how quotes that are routed using this method are printed (both in batch and online).
  - Use **Batch Control** to define the process that creates the flat file that is passed to your quote printing software. Refer to Technical Implementation of Printing Quotes In Batch for more information about these processes.
  - Use **Extract Algorithm** to define the plug-in that constructs the "flat file records" that contain the information merged onto quotes routed using this method. Refer to Printing Quotes for more information.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference CI_QTE_RTE_TYPE.

Setting Up Terms and Conditions

As described under Legal Terms and Conditions May Be Specified On SAs, your SA type start options can reference terms and conditions (T&C's) that should be defaulted onto new service agreements (both real and proposal). Each T&C is identified with a terms and condition code. To define a terms and conditions code, open **Admin** > **Sales & Marketing** > **Terms and Conditions** > **Add.**

➤

NOTE:
T&C print order. The value of the T&C code controls the order in which the T&C appears on the printed quote. This means you should assign these codes in some type of structured format (e.g., 01...) if you would like them to appear in a certain order.

Description of Page

Enter a unique **Terms and Condition** code and **Description**. Use **Text** to describe the exact terms.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference CI_TC.

Setting Up Decline Reasons

A proposal SA decline reason must be supplied when a proposal SA is declined. Open **Admin** > **Sales & Marketing** > **Proposal SA Decline Reason** to define your reason codes.

Description of Page

Enter an easily recognizable **Decline Reason** and **Description** for each proposal SA decline reason.

Where Used

Decline reasons are used when a quote detail is declined.

Setting Up Customer Classes For Quotes

An optional plug-in spot exists on customer class where you can introduce additional logic to be executed when a quote is completed for an account that belongs to this customer class. The base package comes supplied with a sample algorithm that creates a workflow process when a quote is completed (refer to the algorithm QTEC-WP for more information).

Defining Case Management Options

Case management functionality is a highly configurable tool your organization can use to manage many situations, including (but certainly not limited to) the following:

- a high-bill complaint,
- a bankruptcy,
- an inspection of a premise,
- a customer's request for literature,
- an application for new service,
- a contractor's request to extend a line,
- a customer's rejection of a quote,
- a customer's request to change service providers on a future date,
- the processing of a market message in a deregulated environment,
- ... (the list is only limited by your time and imagination)

Obviously the steps involved in the resolution of the above cases are very different. The topics in this section describe how to configure the system to manage your cases as per your organization's desires.

➤ **NOTE:**
Separate module. The Case Management functionality is associated with a separate module. If the Case Management module is not applicable to your business you may turn it off. Refer to Turn Off A Function Module for more information.

● **FASTPATH:**
Refer to Case Management for a description of how end-users use cases.

The Big Picture Of Cases

The topics in this section provide background information about how to configure the system to support your case management requirements.

Case Type Controls Everything

Whenever a user creates a case, they must specify the type of case (e.g., high-bill complaint, literature request, etc.). The case type controls how the case is handled.

Case types hold the business rules that control cases. Since these business rules can sometimes be quite complicated, setting up case types requires planning and foresight. The topics in this section describe the type of business rules that can be configured on your case types.

Person / Account / Premise Applicability

Some types of cases may be person-oriented, others may be premise-oriented, and still others may be account-oriented. For example:

- Cases used to keep track of a literature request would reference the person who requested the literature.
- Cases used to keep track of the inspection of a property would reference the premise being inspected.
- Cases used to keep track of a high-bill complaint would reference the account associated with this bill(s) being disputed.

When you set up a case type, you define if its cases must reference a person, account, and/or premise. Note, any combination of these objects is permitted on a case.

Contact Information Applicability

When a case is created, you may want to keep track of how to contact its originator. For example, you may want to record the originator's email address or phone number. When you set up a case type, you define if contact information is required, optional or not allowed on its cases.

Business Object Association

A case type may reference a Business Object, which serves as a link between cases of that type and the options that are associated with the business object.

Additional Information

Some of your cases may require additional information (in the form of characteristics). For example, a high-bill complaint may require at least one bill. When you set up a case type, you can define the additional fields that are required. In addition, you can define default values for these fields.

The case functionality also allows you to require characteristics when a case enters a given state. Refer to Required Fields Before A Case Enters A State for the details.

➤ **NOTE:**
Requiring supporting documents. Because any type of characteristic can be referenced on a case, you can require references to supporting documents by requiring a file location characteristic.

Access Rights

You can take advantage of the system's security to restrict cases of a given type to certain users. For example, you can restrict high-bill complaints to specific user groups.

The following points describe how to implement this type of security:

- Create an application service for each type of case you need to secure
- Define the access modes Add, Inquire and Change for each application service
- Define the applicable application service on each case type
- Link the appropriate user groups to each application service
    - For user groups that are allowed to add cases of a given type, define Add as a valid access mode.
    - For user groups that are allowed to view cases of a given type, define Inquire as a valid access mode
    - For user groups that are allowed to change cases of a given type, define Change as a valid access mode

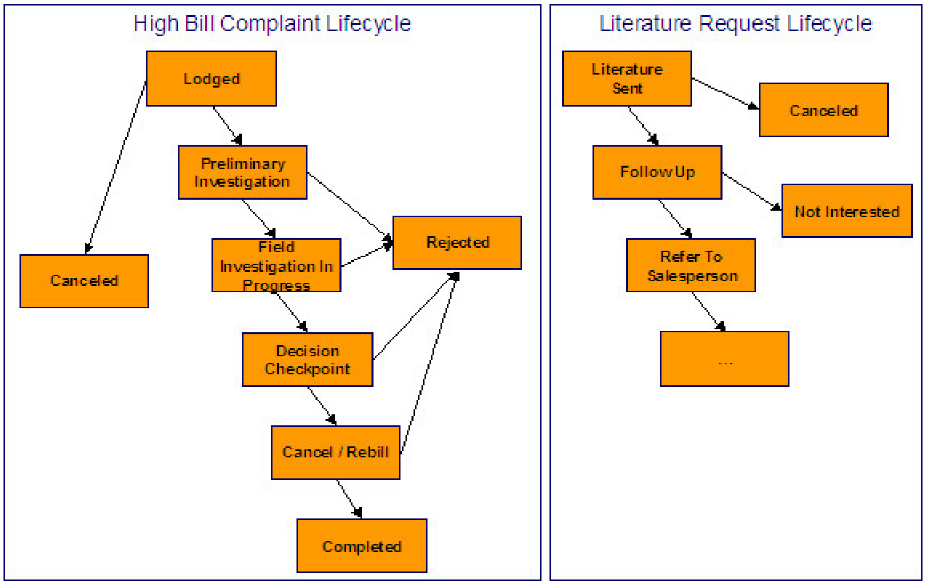
If you restrict access to a case type's cases, you can further restrict which users can work on cases given the status of the case. Refer to Which Users Can Transition A Case for more information.

NOTE:
Restricting access to cases is optional. If you don't specify an application service on a case type, all users (who have access to the case transaction) may access its cases.

Lifecycle

Many objects in the system have predefined lifecycles whose rules are governed by the base-package and cannot be changed. For example, a service agreement starts out in the Pending Start state and eventually becomes Closed when it's been final billed (and paid). You can't change the system to allow a service agreement to starts its life in the Closed state.

The lifecycle of cases is not governed by the base package. Rather, you define the lifecycle of your cases when you set up their case types. Examine the following diagrams; the one on the left shows the potential lifecycle of a case that manages a high-bill complaint, the one on the right shows the potential lifecycle of a case that manages a customer's literature request.



Potential Lifecycles Of Two Types Of Cases

NOTE:

> **Just examples.** The above lifecycles are just examples. When you set up your case types, you must define the valid states for your case type.

The topics that follow describe important concepts that are illustrated in the above diagrams.

Valid States versus State Transition Rules

The orange boxes in the above diagram show the potential valid states a given case can have. The lines between the boxes indicate the state transition rules. These rules govern the states a case can move to while it's 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 case type, you define both its valid states and the state transition rules.

Transitory States

You can define a state in a case type as **Transitory** if you do not wish the case to exist in a particular state. For example, let's assume that an algorithm is associated with the Decision Checkpoint state (Enter Processing) that would automatically determine the next state for the case (i.e. Cancel/Rebill or Reject) and also contains logic to transition the case accordingly. In this scenario, you may not ever want the case to exist in the Decision Checkpoint state, so that a user won't ever see a high bill complaint in that state. If the other states were marked as non-transitory, and an error were to occur during the transition from Decision Checkpoint to Cancel/Rebill, the case would roll back any changes to data made in the Cancel/Rebill (Enter Processing) state along with the changes made in the Decision Checkpoint state, and would end up in the Field Investigation In Progress state - the last non-transitory state prior to Decision Checkpoint.

One Initial State and Multiple Final States

When you set up a case type's states, you must pick one as the initial state. The initial state is the state assigned to new cases of a given type. For example, high-bill complaint cases have an initial state of Lodged, whereas literature request cases have an initial state of Literature Sent.

You must also define which statuses are considered to be "final". When a case enters a "final" state, it is complete and no further action is necessary. You might want to think of the "final" states as the potential outcomes of a case. For example, a high-bill complaint has potential outcomes of Completed, Rejected, and Canceled.

The "final" states are used by the system to differentiate between open and closed cases. For example, an alert highlights when the person / account / premise in context has open cases (this alert only exists if you've plugged-in the appropriate installation alert).

Allowing A Case To Be Reopened

You can set up your state transition rules to allow a case 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 a case type accordingly.

Make Sure To Have A Canceled State

The system does not allow you to delete a case. Therefore, if you want to support logical deletion, you should have a status of Canceled early in a case type's lifecycle. Doing this allows a user to cancel (i.e., logically delete) a case.

➤

NOTE:

Cancel reason. You might want to consider setting up your case types to require a cancel reason (in the form of a predefined value characteristic) when a user cancels a case. Refer to Required Fields Before A Case Enters A State for more information.

Buttons Are Used To Transition A Case From Status To Status

When a case is displayed on Case - Main, a separate button is shown for each state into which the case can be transitioned. For example, a high-bill complaint case that is in the Lodged state would show two buttons: **Start Investigation** and **Cancel**. If the user presses the **Start Investigation** button, the case is transitioned to the Preliminary Investigation state. If the user presses the **Cancel** button, the case is moved to the Canceled state.

You may define the text displayed on the button differently for each state transition. This allows the action description to be varied according to the previous status. For example, the button to transition from New to Active may be labeled **Activate**, but the button to change from Closed to Active may be labeled **Reactivate**.

Refer to Which Users Can Transition A Case for instructions describing how to restrict users to specific actions.

State Transitions Are Audited

The system maintains an audit trail whenever a case transitions from one state to another. This audit is shown in the case's log.

Reports and Analytics Highlight Productivity

When you set up a case type's lifecycle, keep in mind that several reports and analytics highlight how long it took cases to transition into a state. For example, you can use a report to see how long it took high-bill complaints to be completed (or initially actioned or ...). Refer to the Reports chapter for the details of case reports.

Status-Specific Business Rules

As described in Lifecycle, when you set up a case type, you define the possible states its cases can pass through. The following topics describe business rules that can be configured for each state.

A Script That Helps A User Work Through A Case

You can define a Business Process Assistant script that helps a user work a case while it's in a given state. For example, when you set up the Preliminary Investigation state for the high-bill complaint case type, you can define a script. A user can then easily launch this script to help them work through a case in this state.

Please keep the following in mind when you're designing how to integrate BPA scripts with your cases:

- You can have a different script for each state. For example, you could develop a script to help a user work on a case while it's in the Preliminary Investigation state and a different script to help them work in a case while it's in the Decision Checkpoint state.
- Rather than make a user launch a script by pressing a hyperlink on the case page, you can have the system automatically launch the script while the case is in a given state. Refer to Script Launching Option for more information.
- You can also have the system automatically launch a script when a user selects a To Do entry. Refer to Launching Scripts When To Do Entries Are Selected for more information.

FASTPATH:
Refer to Scripts and Cases for more information about how to streamline your case processing with scripts.

Required Fields Before A Case Enters A State

You can define additional fields (i.e., characteristics) that are required before a case can enter a given state. For example,

- You can indicate a high-bill complaint must reference at least one bill before it enters the Preliminary Investigation state.
- You can indicate a case must reference a cancel reason before it enters the Canceled state.

You do this by indicating that characteristics (that were optional when the case was added) are required when a case enters a given state.

Validation Before A Case Enters A State

You can define validation that executes before a case can enter a given state. For example, you can indicate the case must have been assigned a responsible user before it can enter the Preliminary Investigation state. This validation logic is held in algorithms that are plugged in on the case type and therefore you can define any type of validation.

Additional Processing When Entering A State

You can define additional processing that should happen when a case enters a given state. For example, you can have a letter created when a high-bill complaint case is Rejected. Similarly, you can have a To Do entry created when a high bill complaint enters the Preliminary Investigation state. This additional processing is held in algorithms that are plugged in on the case type and therefore you can define any type of additional processing.

You can also incorporate state transitioning logic within routines that are executed when a case enters a state, so that you do not need to rely upon CASETRAN to transition your cases. For example, when the state entry routines of the Preliminary Investigation status for a high-bill complaint are executed, they may be designed to transition the case into either the Rejected or Field Investigation In Progress state without waiting. Note that your Exit Validation and Exit Processing logic, if configured for the case state, will still be executed as part of the state transition. Auto-Transition logic for this state will be ignored during this transition.

Validation Before A Case Exits A State

You can define validation that executes before a case can exit a given state. For example, you might want to check the account's balance is less than a given value before a case can exit a given state. This validation logic is held in algorithms that are plugged in on the case type and therefore you can define any type of validation.

Additional Processing When Exiting A State

You can define additional processing that should happen when a case exits a given state. For example, you can have a To Do entry automatically completed when a high bill complaint leaves the Decision Checkpoint state. This additional processing is held in algorithms that are plugged in on the case type and therefore you can define any type of additional processing.

Automatic Transition Rules

You can define rules that automatically transition a case into a different state. For example, you can indicate a literature request should be transitioned to the Follow Up state 1 week after the literature is sent. Similarly, you can indicate a high-bill complaint should transition to the Decision Checkpoint state after the fieldwork is complete. These rules are held in algorithms that are plugged in on the case type and therefore you can define any type of automatic transition rules.

Cases in a state with automatic transition rules are monitored by the CASETRAN background process. Each time this program runs, the respective automatic transition plug-in is called for each such case and it transitions the case if the condition applies.

➤ NOTE:
When to execute CASETRAN. Because your automatic transition rules will be dependent on your business requirements, you need to think carefully about when you run the CASETRAN background process. For example, if you have automatic transition rules that transition a case to a new state when a related field activity is completed, you would want to schedule this job to run after field activities are uploaded. If you have rules to transition a case after a customer pays a deposit, you'd want to schedule this job to run after payments are uploaded. Bottom line - your business rules will dictate the frequency of execution.

When the user adds a new case or changes the state of a case manually the system attempts to auto-transition the case to subsequent statuses as necessary. If auto-transition rules apply to the new state (and to subsequent ones) they would be executed right away. In other words, you don't need to wait for the auto-transition background process to be executed. An indication that the case was auto-transitioned online is displayed right below the action buttons section.

➤ NOTE:
Auto-Transition Errors. Online auto-transition is performed recursively committing each successful state transition to the database. It is performed up to 100 times or until an error is encountered during the process. If this happens, auto-transition stops at the last **non-transitory** state into which a successful transition had occurred. Two case log entries will be generated automatically - one containing the message that a transition error has occurred, and a second containing the actual error message. A To Do entry will also be generated automatically upon rollback. The type of this To Do entry will be taken from 1) the Case Transition Exception To Do Type **option** for the **Business Object** associated with the case type, and if this is

not populated, 2) the Exception To Do Type indicated on the Case Options Feature Configuration. All of the above error handling is true for both batch and online processing of cases.

➤ **NOTE:**

Triggering Auto-Transition. If you have a customized process that affects the state of a case and you want the case to be auto-transitioned right away, i.e. not wait for the next scheduled CASETRAN background process to execute, you can customize that process to trigger auto-transition for the specific case, or you can put the state transition logic into the routines that execute at state entry time.

Script Launching Option

You can define whether the script associated with a given state is to be automatically launched while the case is in that state. The system supports the following options:

- Launch the script only if no script is currently active.
- Always launch the script unless this specific script is currently active.

  WARNING:

  With this option, if a script is currently open in the page's BPA script area then it will be automatically closed and the case script will open.

- Do not automatically launch the script.

You do this by plugging-in a Script Launching algorithm for the given state. If no such plug-in is provided the script is not automatically launched.

Which Users Can Transition A Case Into A State

If you have restricted access to a case type, you can further restrict which user groups are allowed to transition a case into specific states. For example, you can control which user group can transition a high bill complaint into the Preliminary Investigation state. The following points describe how this is done:

- Define actions on the application service defined on the case type. You must define an action for each status that you need to secure.
- Define each status's corresponding action. Note, you only need to link a status to an action if it's secured. Any user with access to the case type can perform statuses that aren't linked to actions.
- Define the transition role for each status's valid next status. You can assign valid next statuses to be reachable via system (only), or system and user.
- Define which user groups have access to the actions (i.e., statuses). In addition, these user groups should have access to the Change action.

Responsible User Applicability

Some of your cases may require a "responsible user". This is the user who has overall responsibility for the case. When you set up a case type, you define if a responsible user is required, optional or not allowed on its cases.

The following points describe how to set up the system if a responsible user is not required when a case is first created, but is later in its lifecycle:

- Indicate that a responsible user is optional on the case type
- Plug-in either an exit validation or entry validation algorithm on one of the case type's states to require a responsible user at some point in a case's lifecycle

➤ **NOTE:**
Address To Do entries to the responsible user. If you use the base-package algorithm to create a To Do entry when a case enters a given state, you can indicate that the To Do entry should be addressed to the responsible user on the case.

Scripts and Cases

There are three ways Business Process Assistant scripts can be used to manage cases:

- You can create a BPA script to help users create a case. For example, a script can help a user create a new high-bill complaint.
- Using a script to create a case can save a user a lot of time (and training efforts). This is because the script can automatically populate many fields on the case based on answers to questions.

Refer to Initiating Scripts for a description of how end-users initiate scripts.

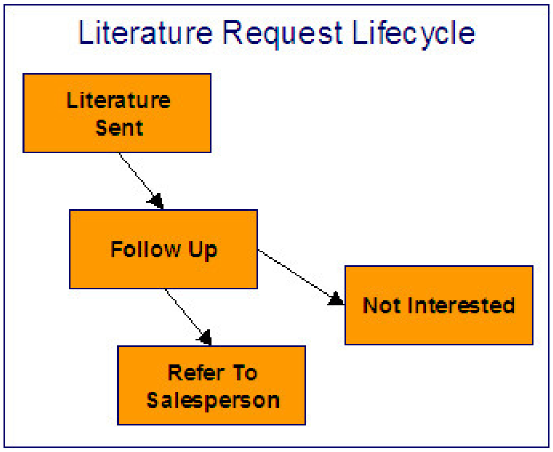
- You can create a script to help users work on a case when it's in a given state. Refer to A Script That Helps A User Work A Case for more information.
- You can set up your case types to create To Do entries to notify users when cases exist that require their attention. Users can complete many of these ToDo entries without assistance. However, you can set up the system to automatically launch a script when a user selects a ToDo entry. For example, consider a ToDo entry that highlights a high-bill complaint that requires investigation. You can set up the system to execute a specific script when a user selects this ToDo entry. This script might guide the user through the investigation process (and help them update the case). Refer to Executing A Script When A To Do Entry Is Selected for more information.

To Do's and Cases

The topics in this section provide background information about how to facilitate case management with To Do entries.

Creating and Completing To Do Entries

You can configure your case types to create and complete To Do entries when a case enters or exits a state. Let's use the following lifecycle diagram to illustrate a potential use of To Do's:



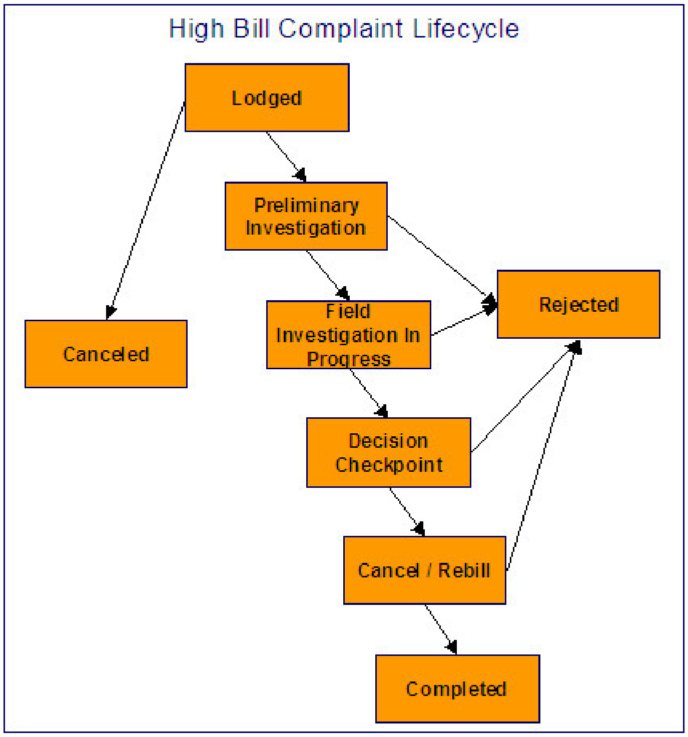
Let's assume the following:

- You want a To Do entry created when a literature request case enters the Follow Up state. You want this To Do automatically completed when the case enters either the Refer To Salesperson or Not Interested states. Note, we refer to this as the "first" To Do entry below.
- You want a different To Do entry created when a case enters the Refer To Salesperson state. You do not want the system to automatically complete this entry (the sales person must manually do this). Note, we refer to this as the "second" To Do entry below.

To implement the above, you would set up a case type as follows:

- Plug-in an entry processing algorithm on the Follow Up status to create the first To Do entry.
- Plug-in an exit processing algorithm on the Follow Up status to complete the first To Do entry.
- Plug-in an entry processing algorithm on the Refer To Salesperson status to create the second To Do entry.

While the case type illustrated above had a single To Do entry "active" at any point in time, you can easily configure a case type to have multiple To Do entries active at any point in time. Let's use the following lifecycle diagram to illustrate this point:


High Bill Complaint Lifecycle
Lodged
Preliminary Investigation
Rejected
Canceled
Field Investigation In Progress
Decision Checkpoint
Cancel / Rebill
Completed

Let's assume the following:

- You want a To Do entry created when a high bill complaint is created and you want it completed when the case reaches the Canceled, Rejected or Approved states. This To Do entry could be used by a supervisor to monitor the number of high-bill complaints being worked. Note, we refer to this as the "first" To Do entry below.
- You want a different To Do entry created when the case enters the Preliminary Investigation state and you want this entry automatically completed when the case leaves this state. Note, we refer to this as the "second" To Do entry below.
- You want a different To Do entry created when the case enters the Decision Checkpoint state and you want this entry automatically completed when the case leaves this state. Note, we refer to this as the "third" To Do entry below.

To implement the above, you would set up the case type as follows:

- Plug-in an entry processing algorithm on the Lodged status to create the first To Do entry. Plug-in an entry processing algorithm on the Canceled, Rejected and Completed statuses to complete this entry.
- Plug-in an entry processing algorithm on the Preliminary Investigation status to create the second To Do entry. Plug-in an exit processing algorithm on the Preliminary Investigation status to complete this entry. We elected to use an exit processing algorithm because we only have to plug it in on one status. If we'd used an entry processing algorithm, we would need to plug it in on the 2 statuses into which a Preliminary Investigation status can transition.
- Plug-in an entry processing algorithm on the Decision Checkpoint status to create the third To Do entry. Plug-in an exit processing algorithm on the Decision Checkpoint status to complete this entry.

Launching Scripts When To Do Entries Are Selected

You can set up your case types to create To Do entries to notify users when cases exist that require their attention. Users can complete many of these To Do entries 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 high-bill complaint that requires investigation. You can set up the system to execute a specific script when a user selects this type of To Do entry. This script might guide the user through the investigation process. Refer to Executing A Script When A To Do Entry Is Selected for more information.

All To Do Entries Are Visible

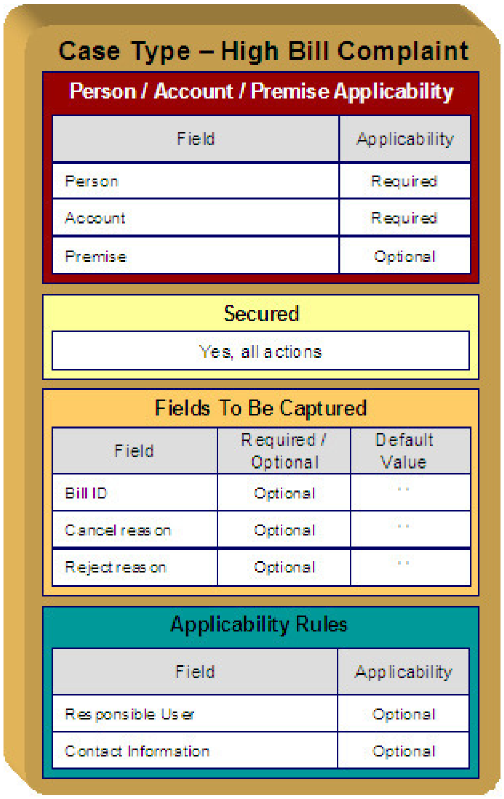
When a case is displayed on Case Maintenance, the system summarizes the number of To Do entries associated with the case (if you've set up your To Do types appropriately).

Examples of Case Types

The topics that follow provide examples of case types related to several business processes. Use the information in this section to form an intuitive understanding of case types. After attaining this understanding, you'll be ready to design your own case types.

High Bill Complaint

Some organizations will set up a case to manage a high-bill complaint. The following diagram illustrates how such a case type might look:



Case Type – High Bill Complaint

Person / Account / Premise Applicability

Field	Applicability
Person	Required
Account	Required
Premise	Optional

Secured

Yes, all actions

Fields To Be Captured

Field	Required / Optional	Default Value
Bill ID	Optional	' '
Cancel reason	Optional	' '
Reject reason	Optional	' '

Applicability Rules

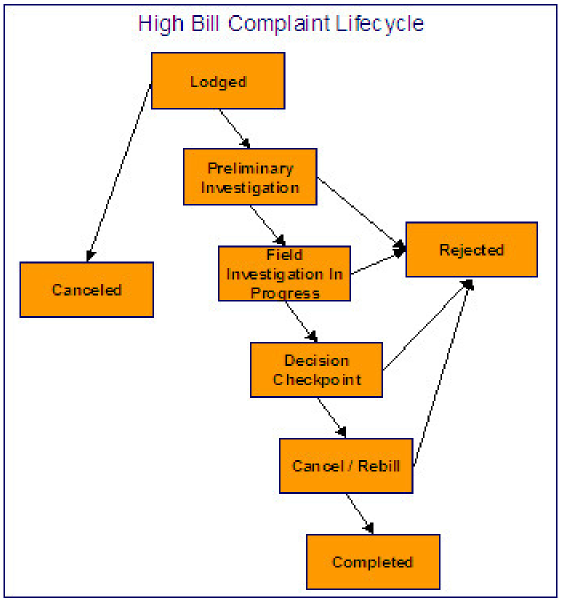
Field	Applicability
Responsible User	Optional
Contact Information	Optional

Note the following about the High Bill Complaint case type:

- Notice that we set up the case type to require a person and an account, but premise is optional. This is because a bill can span multiple premises and knowing the premise isn't so important on cases of this type.
- We need to restrict high-bill complaints to specific user groups. This means we need to set up a specific application service for this case type (that has a separate "action" for each status).

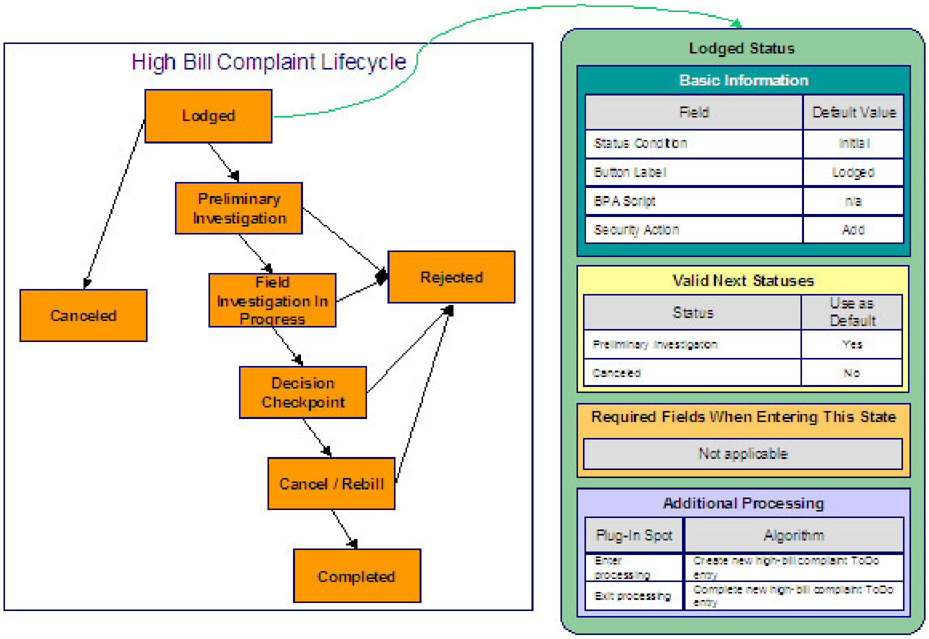
- Cases of this type have 3 additional fields over their lifetime. Notice the following:
  - The **Bill ID** characteristic is set up to be optional. This is because we've assumed that sometimes a high-bill complaint case can be lodged when you can't find the bill in question and you still want to log the case.
  - Later in this section, you'll see that we've configured the Preliminary Investigation status to require a **Bill ID** before a case can enter this state.
  - Both the **High Bill Complaint Cancel Reason** and **Reject Reason** are optional. Later in this section, you'll see that we've configured the Canceled and Rejected statuses to require these fields, respectively.
- Cases of this type do not need a **Responsible User** when first created. Later in this section, you'll see how we've configured the Preliminary Investigation status to require a **Responsible User** before a case can enter this state.
- Cases of this type do not need **Contact Information**. This was a subjective decision and depends on your organization's philosophy.

The topics that follow describe each of the statuses in a high-bill complaint's lifecycle. We have assumed the following state transition rules:


High Bill Complaint Lifecycle
Lodged
Preliminary Investigation
Canceled
Field Investigation In Progress
Rejected
Decision Checkpoint
Cancel / Rebill
Completed

Lodged High Bill Complaint

The following is an example of the configuration of the Lodged status for high bill complaint cases.


High Bill Complaint Lifecycle
Lodged
Preliminary Investigation
Canceled
Field Investigation In Progress
Rejected
Decision Checkpoint
Cancel / Rebill
Completed

Lodged Status

Basic Information

Field	Default Value
Status Condition	Initial
Button Label	Lodged
BPA Script	n/a
Security Action	Add

Valid Next Statuses

Status	Use as Default
Preliminary Investigation	Yes
Canceled	No

Required Fields When Entering This State

Not applicable

Additional Processing

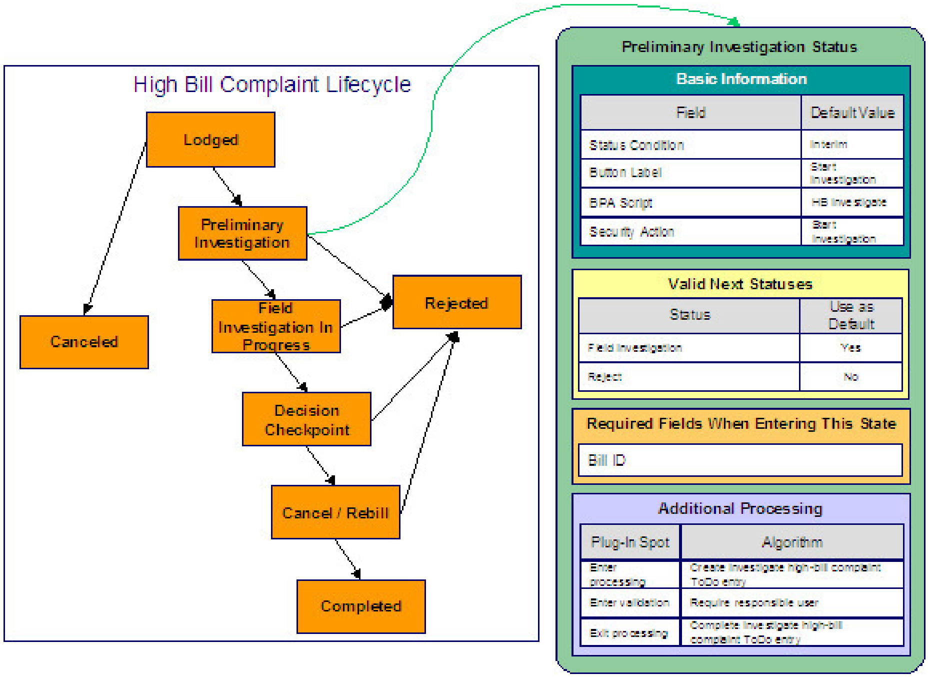
Plug-In Spot	Algorithm
Enter processing	Create new high-bill complaint ToDo entry
Exit processing	Complete new high-bill complaint ToDo entry

Note the following about this status:

- Lodged is the initial state. The initial state is the state populated on new cases of a given type. Remember that only one status can be defined as the initial state.
- It has a button label even though it's the initial state. The above diagram doesn't allow a user to ever transition a case into this state and therefore there will never be a button with such a label. However, it's a required field just in case you change your business rules.
- We have decided not to use a BPA script to help a user work on a high-bill complaint when it's in this state (this is probably not the best decision as BPA scripts can be quite useful).
- We have associated the Add action with this status. This means that only users with rights to the add action for the application service defined on the case type can add cases of this type.
- Notice that Valid Next Statuses are what restricts a case in this state to be transitioned to the Canceled and Preliminary Investigation states.
- Notice that the Additional Processing plug-ins create and complete a To Do entry when a case enters and exits this state, respectively.

Preliminary Investigation High Bill Complaint

The following is an example of the configuration of the Preliminary Investigation status for high bill complaint cases.


High Bill Complaint Lifecycle
Lodged
Preliminary Investigation
Canceled
Field Investigation In Progress
Rejected
Decision Checkpoint
Cancel / Rebill
Completed

Preliminary Investigation Status

Basic Information

Field	Default Value
Status Condition	Interim
Button Label	Start Investigation
BPA Script	HB Investigate
Security Action	Start Investigation

Valid Next Statuses

Status	Use as Default
Field Investigation	Yes
Reject	No

Required Fields When Entering This State

Bill ID

Additional Processing

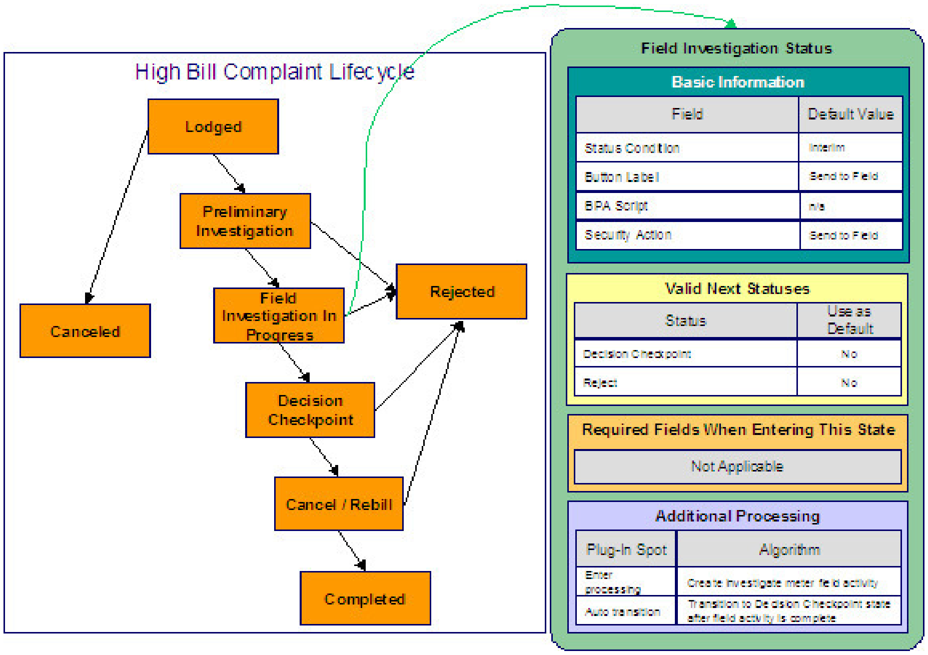
Plug-In Spot	Algorithm
Enter processing	Create investigate high-bill complaint ToDo entry
Enter validation	Require responsible user
Exit processing	Complete investigate high-bill complaint ToDo entry

Note the following about this status:

- Preliminary Investigation is an interim state (meaning that it's not an initial or final state).
- It has a button label of **Start Investigation**. This is the label on the button that a user presses to move a case into this state. This button will only appear on cases that are in the Lodged state as this is the only state that can transition into the Preliminary Investigation state.
- We have decided not to specify a BPA script on this status. Rather, we're going to set up the To Do type used to highlight cases in this state to automatically launch an appropriate BPA script when a user selects the To Do entry.
- We have associated the Start Investigation action with this status. This means that only users with rights to this action for the application service defined on the case type can move a case into this state.
- Notice that Valid Next Statuses are what restricts a case in this state to be transitioned to the Field Investigation and Rejected states.
- Notice that a **Bill ID** must be specified on the case before it can be moved into this state.
- Notice that the Additional Processing plug-ins do the following:
  - Create and complete a To Do entry when a case enters and exits this state, respectively
  - Require a responsible user before a case can enter this state

Field Investigation High Bill Complaint

The following is an example of the configuration of the Field Investigation In Progress status for high bill complaint cases.


High Bill Complaint Lifecycle
Lodged
Preliminary Investigation
Canceled
Field Investigation In Progress
Rejected
Decision Checkpoint
Cancel / Rebill
Completed

Field Investigation Status

Basic Information

Field	Default Value
Status Condition	Interim
Button Label	Send to Field
BPA Script	n/a
Security Action	Send to Field

Valid Next Statuses

Status	Use as Default
Decision Checkpoint	No
Reject	No

Required Fields When Entering This State

Not Applicable

Additional Processing

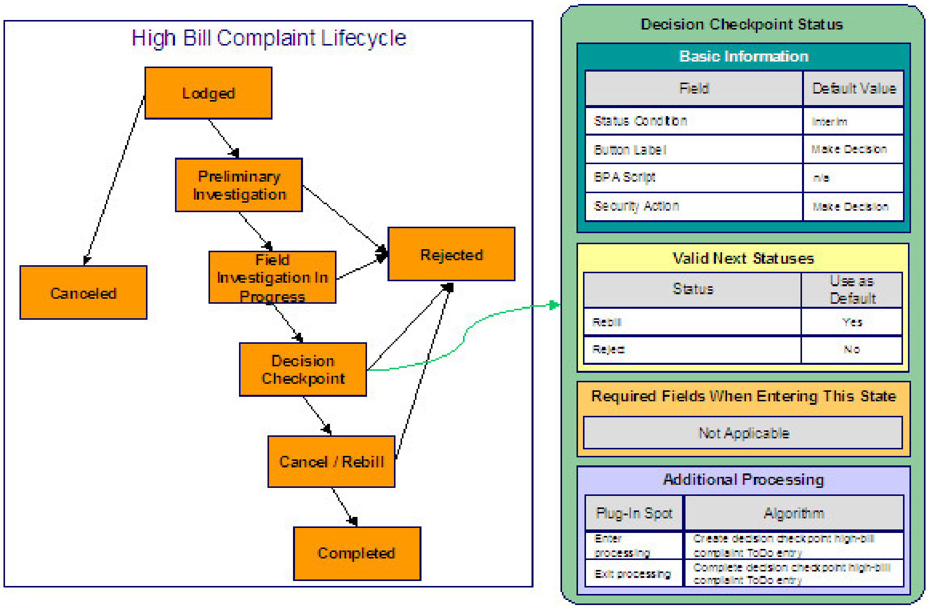
Plug-In Spot	Algorithm
Enter processing	Create investigate meter field activity
Auto transition	Transition to Decision Checkpoint state after field activity is complete

Note the following about this status:

- Field Investigation In Progress is an interim state (meaning that it's not an initial or final state).
- It has a button label of **Send to Field**. This is the label on the button that a user presses to move a case into this state. This button will only appear on cases that are in the Preliminary Investigation state as this is the only state that can transition into the Field Investigation in Progress state.
- We have decided not to specify a BPA script on this status because users don't work cases in this state (see the Additional Processing notes below for why this is the case).
- We have associated the Send to Field action with this status. This means that only users with rights to this action for the application service defined on the case type can move a case into this state.
- Notice that Valid Next Statuses are what restricts a case in this state to be transitioned to the Decision Checkpoint and Rejected states.
- Notice that the Additional Processing plug-ins do the following:
  - Create a field activity when a case enters this state
  - Cause the case to automatically transition to the Decision Checkpoint state when the field activity is completed

Decision Checkpoint High Bill Complaint

The following is an example of the configuration of the Decision Checkpoint status for high bill complaint cases.


High Bill Complaint Lifecycle
Lodged
Preliminary Investigation
Canceled
Field Investigation In Progress
Rejected
Decision Checkpoint
Cancel / Rebill
Completed

Decision Checkpoint Status

Basic Information

Field	Default Value
Status Condition	Interim
Button Label	Make Decision
BPA Script	n/a
Security Action	Make Decision

Valid Next Statuses

Status	Use as Default
Rebill	Yes
Reject	No

Required Fields When Entering This State

Not Applicable

Additional Processing

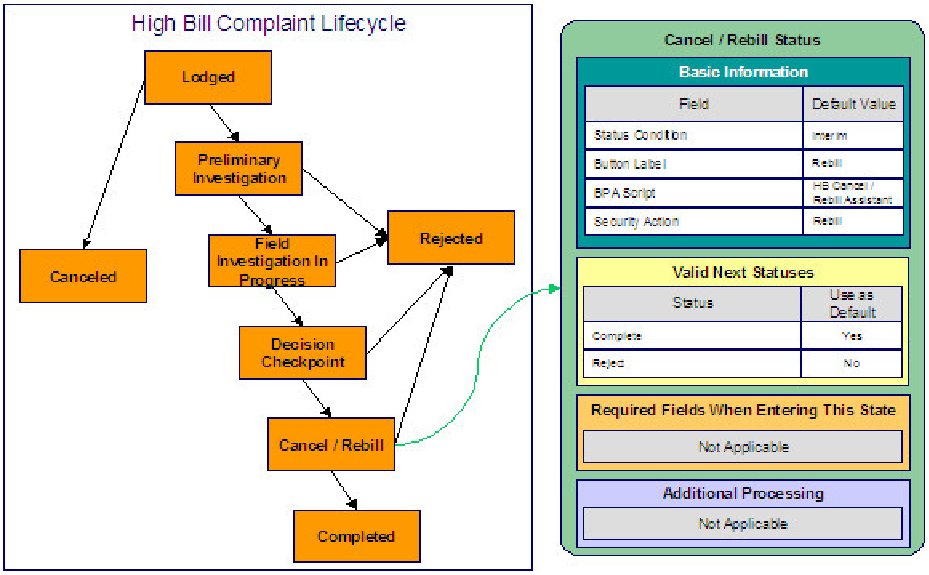
Plug-In Spot	Algorithm
Enter processing	Create decision checkpoint high-bill complaint ToDo entry
Exit processing	Complete decision checkpoint high-bill complaint ToDo entry

Note the following about this status:

- Decision Checkpoint is an interim state (meaning that it's not an initial or final state).
- It has a button label of **Make Decision**. This is the label on the button that a user presses to move a case into this state. This button will only appear on cases that are in the Field Investigation in Progress state as this is the only state that can transition into the Decision Checkpoint state.
- We have decided not to specify a BPA script on this status. Rather, we're going to set up the To Do type used to highlight cases in this state to automatically launch an appropriate BPA script when a user selects the To Do entry.
- We have associated the Make Decision action with this status. This means that only users with rights to this action for the application service defined on the case type can move a case into this state. Because the system will automatically transition cases into this state when the related field activity is complete, users will probably never press this button (and you may wish to prevent users from pressing this button be restricting security rights to the related action).
- Notice that Valid Next Statuses are what restricts a case in this state to be transitioned to the Cancel / Rebill and Rejected states.
- Notice that the Additional Processing plug-ins create and complete a To Do entry when a case enters and exits this state, respectively.

Cancel Rebill High Bill Complaint

The following is an example of the configuration of the Cancel / Rebill status for high bill complaint cases.


High Bill Complaint Lifecycle
Lodged
Preliminary Investigation
Canceled
Field Investigation In Progress
Rejected
Decision Checkpoint
Cancel / Rebill
Completed

Cancel / Rebill Status

Basic Information

Field	Default Value
Status Condition	Interim
Button Label	Rebill
BPA Script	HB Cancel / Rebill Assistant
Security Action	Rebill

Valid Next Statuses

Status	Use as Default
Complete	Yes
Reject	No

Required Fields When Entering This State

Not Applicable

Additional Processing

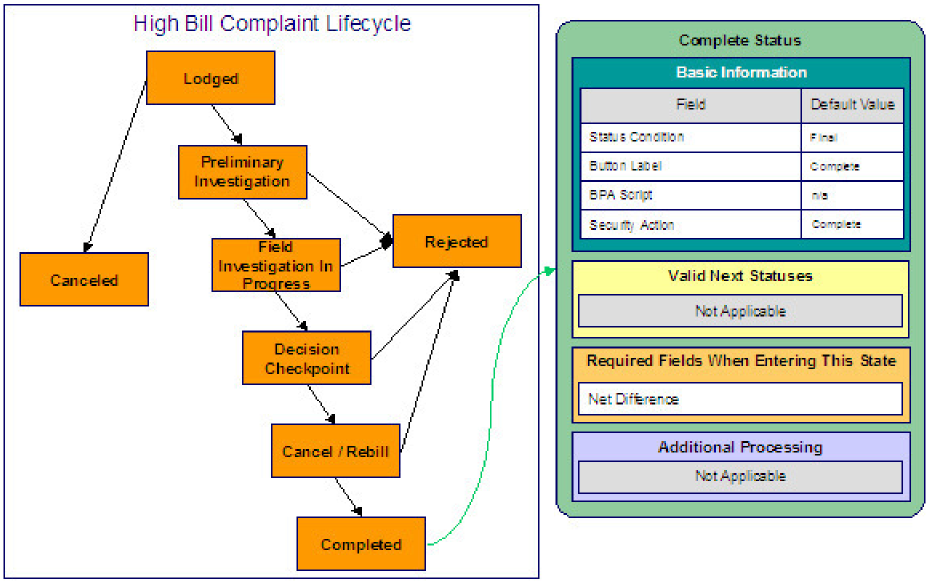
Not Applicable

Note the following about this status:

- Cancel / Rebill is an interim state (meaning that it's not an initial or final state).
- It has a button label of **Cancel / Rebill**. This is the label on the button that a user presses to move a case into this state. This button will only appear on cases that are in the Decision Checkpoint state as this is the only state that can transition into the Cancel / Rebill state.
- We have referenced a BPA script that can assist a user in the cancel / rebill efforts.
- We have associated the Rebill action with this status. This means that only users with rights to this action for the application service defined on the case type can move a case into this state.
- Notice that Valid Next Statuses are what restricts a case in this state to be transitioned to the Completed and Rejected states.

Completed High Bill Complaint

The following is an example of the configuration of the Completed status for high bill complaint cases.



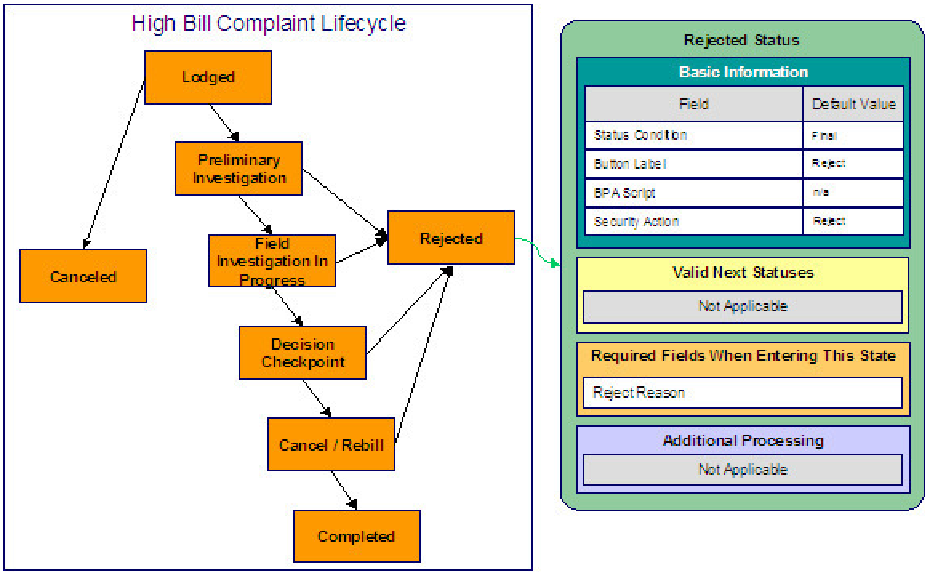
Field	Default Value
Status Condition	Final
Button Label	Complete
BPA Script	n/a
Security Action	Complete

Note the following about this status:

- Completed is a final state (meaning that no further action is necessary).
- It has a button label of **Complete**. This is the label on the button that a user presses to move a case into this state. This button will only appear on cases that are in the Cancel / Rebill state as this is the only state that can transition into the Completed state.
- We have not referenced a BPA script because this is a final state (and no additional user action is necessary).
- We have associated the Complete action with this status. This means that only users with rights to this action for the application service defined on the case type can move a case into this state.
- Notice that there are no Valid Next Statuses (because this is a final state). If you wanted to allow a Completed case to be reopened, you would need to define the state into which a Completed case could be transitioned.
- Notice that the **Net Difference** must be specified on the case before it can be moved into this state. This would be the difference to the customer's balance after the cancel rebill took place.

Rejected High Bill Complaint

The following is an example of the configuration of the Rejected status for high bill complaint cases.


High Bill Complaint Lifecycle
Lodged
Preliminary Investigation
Canceled
Field Investigation In Progress
Rejected
Decision Checkpoint
Cancel / Rebill
Completed

Rejected Status

Basic Information

Field	Default Value
Status Condition	Final
Button Label	Reject
BPA Script	n/a
Security Action	Reject

Valid Next Statuses

Not Applicable

Required Fields When Entering This State

Reject Reason

Additional Processing

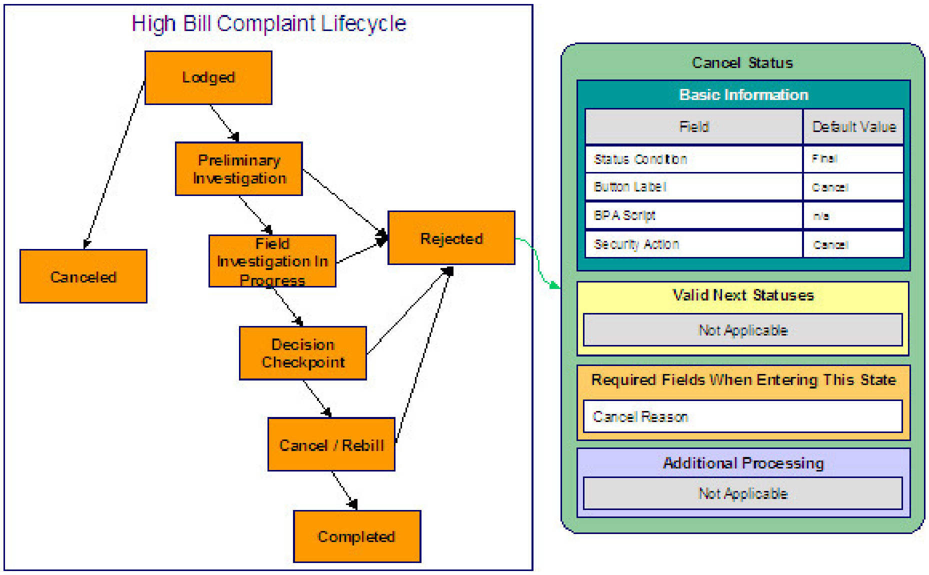
Not Applicable

Note the following about this status:

- Rejected is a final state (meaning that no further action is necessary).
- It has a button label of **Reject**. This is the label on the button that a user presses to move a case into this state. This button will appear on cases that are in the Preliminary Investigation, Field Investigation in Progress , Decision Checkpoint and Cancel / Rebill states.
- We have not referenced a BPA script because this is a final state (and no additional user action is necessary).
- We have associated the Reject action with this status. This means that only users with rights to this action for the application service defined on the case type can move a case into this state.
- Notice that there are no Valid Next Statuses (because this is a final state). If you wanted to allow a Rejected case to be reopened, you would need to define the state into which a Rejected case could be transitioned.
- Notice that a **Reject Reason** must be specified on the case before it can be moved into this state.

Canceled High Bill Complaint

The following is an example of the configuration of the Canceled status for high bill complaint cases.



Field	Default Value
Status Condition	Final
Button Label	Cancel
BPA Script	n/a
Security Action	Cancel

Note the following about this status:

- Canceled is a final state (meaning that no further action is necessary).
- It has a button label of **Cancel**. This is the label on the button that a user presses to move a case into this state. This button will only appear on cases that are in the Lodged state as this is the only state that can transition into the Completed state.
- We have not referenced a BPA script because this is a final state (and no additional user action is necessary).
- We have associated the Cancel action with this status. This means that only users with rights to this action for the application service defined on the case type can move a case into this state.
- Notice that there are no Valid Next Statuses (because this is a final state). If you wanted to allow a Canceled case to be reopened, you would need to define the state into which a Canceled case could be transitioned.
- Notice that a **Cancel Reason** must be specified on the case before it can be moved into this state.

Setting Up Case Management Options

The topics in this section describe how to set up the system to enable case management.

WARNING:
The following topics assume you thoroughly understand the concepts described under The Big Picture Of Cases.

Installation Options

Case Info May Be Formatted By An Algorithm

An algorithm may be plugged in on the installation record to format the standard case information that appears throughout the system. Refer to CSIN-DFLT for an example of this algorithm.

This algorithm may be further overridden by a "Case information" plug-in on the Case Type. Refer to C1-CT-INFO for an example of this algorithm.

Alert Info Is Controlled By An Installation Algorithm

An algorithm that is plugged in on the installation record is responsible for formatting the alerts that highlight if the person / account / premise in context has open cases. Refer to CCAL-CASE for an example of this algorithm.

Setting Up Application Services

As described under Access Rights, you can prevent unauthorized users from accessing cases. The following points describe how to implement this type of security:

- Create an application service for each case type that needs to be secured
- Create an action on the application service for each status you need to secure
- Link the valid user groups to the application service and define which actions they can perform
- Define the application service on the case type
- Define the related action for each status on the case type / status

Setting Up Scripts

As described under Scripts and Cases, BPA scripts can facilitate the creation and working of cases. Refer to the Defining Script Options for instructions describing how to set up scripts.

Setting Up To Do Types

As described under To Do's and Cases, To Do entries can be used to highlight cases that require user attention.

The following points provide a high-level description of how to create (and complete) To Do entries for a case type:

- Create a To Do type for each different type of To Do entry used during a case's lifecycle
  - On the To Do type, think carefully about the roles whose users can work on the entries
  - Also consider if you would like a BPA script launched when a user selects the entry
- Specify the To Do type on the appropriate entry processing or exit processing algorithm
- If you want the system to automatically complete To Do entries, specify the To Do type on the appropriate entry processing or exit processing algorithm

Please be aware that the case maintenance transaction highlights the number of open and being worked To Do entries linked to the case being displayed on the page. However, the system can only do this if the To Do entries reference a foreign-key characteristic whose foreign key references the case table. If you use the CSEN-TD algorithm to create To Do entries when a case enters a given state, this algorithm will do this for you if:

- You have set up a foreign-key characteristic type whose foreign key references the case table
- In addition, the characteristic type must reference a characteristic entity of To Do Entry

Setting Up Characteristic Types

As described under Additional Information, some of your cases may require additional information (in the form of characteristics). If this is true, you must set up the characteristic types before setting up the case types.

Refer to Setting Up To Do Types for instructions regarding a characteristic type that must be set up in order for the system to know the To Do entries that are associated with a case.

If you use the CSEN-CC algorithm to create customer contacts when a case enters a given state, you should set up a foreign-key characteristic type as follows:

- Its foreign key must reference the case table
- In addition, the characteristic type must reference a characteristic entity of Customer Contact

Setting Up Case Types

The case type maintenance transaction is used to maintain your case types. The topics in this section describe how to use this transaction.

FASTPATH:
Refer to The Big Picture Of Cases for more information about how a case type encapsulates the business rules that govern a case.

Case Type - Main

Use this page to define basic information about a case type.

Open the case type page by selecting **Admin** > **Customer** > **Case Type** > **Case Type** > **Add.**

NOTE:
Recommendation. Before using this transaction, we strongly recommend that you review the Examples of Case Types.

Main Information

Enter a unique **Case Type** code and **Description** for the case type.

Use **Long Description** to provide a more detailed explanation of the purpose of the case type.

If your implementation requires restricting case types by CIS Division, specify a **CIS Division** on the case type.

Person Usage controls the applicability of a person on cases of this type. Select Required if a person must be defined on this type of case. Select Optional if a person can optionally be defined on this type of case. Select Not Allowed if a person is not allowed on this type of case.

Account Usage controls the applicability of an account on cases of this type. Select Required if an account must be defined on this type of case. Select Optional if an account can optionally be defined on this type of case. Select Not Allowed if an account is not allowed on this type of case.

Premise Usage controls the applicability of a premise on cases of this type. Select Required if a premise must be defined on this type of case. Select Optional if a premise can optionally be defined on this type of case. Select Not Allowed if a premise is not allowed on this type of case.

If you need to restrict access to cases of this type to specific user groups, reference the appropriate **Application Service**. Refer to Setting Up Application Services for the details of how to secure access to your cases.

If you are configuring a case type to handle the processing of data defined via a **Business Object**, associating the case type with a business object serves to link the properties of the business object (e.g. BO options) with cases of that type. Refer to Business Objects for further information. In addition, refer to Automatic Transition Rules for information on the role of BO options in case auto-transition errors.

Responsible User Usage controls the applicability of a responsible user on cases of this type. Select Required if a responsible user must be defined on this type of case. Select Optional if a responsible user can optionally be defined on this type of case. Select Not Allowed if a responsible user is not allowed on this type of case. Refer to Responsible User Applicability for more information.

Contact Information Fields

There are three contact information fields: **Contact Person & Method Usage**, **Contact Instructions Usage**, and **Callback Phone Usage**. These fields are used to determine whether or not each type of contact information must be entered on case records with this case type. Select Required if the contact information must be entered, select Optional if the user can choose whether or not to include the contact information on this type of case, or select Not Allowed if the contact information cannot be entered on this type of case.

Algorithms

The **Algorithms** grid contains algorithms that control functions for cases of this type. 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.

The following table describes each **System Event**.

System Event	Optional / Required	Description
Case Information	Optional	We use the term "Case information" to describe the basic information that appears throughout the system to describe a case. The data that appears in "case information" is constructed using this algorithm. Plug an algorithm into this spot to override the "Case information" algorithm on installation options or the system default "Case information" if no such algorithm is defined on installation options. Click here to see the algorithm types available for this system event.

Case Type Tree

The tree summarizes the case type's lifecycle. You can use the hyperlinks to transfer you to the **Lifecycle** tab with the corresponding status displayed.

Case Type - Case Characteristics

To define characteristics that can be defined for cases of this type, open **Admin** > **Customer** > **Case Type** > **Search** and navigate to the **Case Characteristics** page.

Description of Page

Use **Sequence** to control the order in which characteristics are defaulted. Turn on the **Required** switch if the **Characteristic Type** must be defined on cases of this type. Turn on the **Default** switch to default the **Characteristic Type** when cases of this type are created. Enter a **Characteristic Value** to use as the default for a given **Characteristic Type** when the **Default** box is checked. Refer to Required Fields Before A Case Enters A State for a description of how you can make option characteristics required at later stages in a case's lifecycle.

Case Type - Lifecycle

Case types that involve multiple users and multiple potential outcomes have complex lifecycles. Before you can design a case type's lifecycle, it's important that you thoroughly understand the concepts described under Lifecycle and Status-Specific Business Rules. After thoroughly understanding these concepts, we recommend you perform the following design steps:

- Draw a "state transition diagram" as illustrated above under Lifecycle. Keep in mind that if your state transition diagram is complex, your cases will be complex. While some cases warrant complexity, you should always ask yourself if there aren't better ways to achieve the desired results if your first effort results in complexity.
- Determine which characteristics (if any) are required during each stage of a case's lifecycle
- Determine when To Do entries (if any) should be created (and completed) during a case's lifecycle
- Determine additional validation (if any) that should be executed before a case enters and exits each state
- Determine additional processing (if any) that should transpire when a case enters or exits each state
- Determine if scripts are warranted to help users work the cases and, if so, design the scripts for each applicable state

When the above tasks are complete, you will be ready to set up a case type's lifecycle.

Open the Lifecycle page by selecting **Admin** > **Case Type** > **Case Type** > **Search** and navigate to the **Lifecycle** page.

> **NOTE:**
> You can navigate to a status by clicking on the respective node in the tree on the Main tab. You can also use the hyperlinks in the Next Statuses grid to display a specific status in the accordion.

Main Information

The **Status** accordion contains an entry for every status in the case type's lifecycle.

Use **Status** to define the unique identifier of the status. This is NOT the status's description, it is simply the unique identifier used by the system.

Use **Description** to define the label that appears on the lifecycle accordion as well as the status displayed on the case.

Use **Script** to reference a BPA script that can assist a user work on a case while it's in this status. Refer to A Script That Helps A User Work Through A Case for the details.

Use **Access Mode** to define the action associated with this status. This field is disabled if an application service is not specified on the Main page. Refer to Access Rights for the details of how to use this field to restrict which users can transition a case into this state.

Use **Batch** to specify a batch control that will auto-transition the case. Any case in a status configured with a batch control will be transitioned when the batch job runs (rather than when CASETRAN is executed). For this purpose, batch process C1-CSTRS (**Case Scheduled Transition**) is supplied with base package, which will execute all Exit Status logic for the current status, and Enter Status logic for the destination status. You may choose to create a batch process with your own transition logic.

➤

NOTE:
If you wish to defer transitioning a case in a particular status until the batch process on your case type status is executed, you should not populate an Auto-Transition algorithm on that status. Otherwise, CASETRAN will transition the case according to your Auto-Transition logic.

Use **Comment** to describe the status. This is for your internal documentation requirements.

Use **Sequence** to define the relative order of this status in the tree on the Main page.

Use **Status Condition** to define if this status is an Initial, Interim or Final state. Refer to One Initial State and Multiple Final States for more information about how this field is used.

Use **Transitory State** to indicate whether a case should ever exist in this state. Only Initial or Interim states can have a transitory state value of No.

The **Alert Flag** is used to indicate whether or not an alert should be displayed for customers with cases in the state. (The alert is shown via the base package Installation - Alert algorithm.)

Algorithms

The **Algorithms** grid contains algorithms that control important functions for cases of this type. 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.

The following table describes each **System Event** (note, all system event's are optional and you can define an unlimited number of algorithms for each event).

System Event	Description
Auto Transition	This algorithm is executed to determine if a case that's in this state should be transitioned into another state. Refer to Automatic Transition Rules for the details. Click here to see the algorithm types available for this system event.
Enter Processing	This algorithm holds additional processing that is executed when a case is transitioned into this state. You can also specify state transition logic within Enter Processing routines. Refer to Additional Processing When Entering A State for the details. Click here to see the algorithm types available for this system event.
Enter Validation	This algorithm holds validation logic that executes before a case can enter a given state. Refer to Validation Before A Case Enters A State for the details.

	Click here to see the algorithm types available for this system event.
Exit Processing	This algorithm holds additional processing that is executed when a case is transitioned out of this state. Refer to Additional Processing When Exiting A State for the details. Click here to see the algorithm types available for this system event.
Exit Validation	This algorithm holds validation logic that executes before a case can by transitioned out of a given state. Refer to Validation Before A Case Exits A State for the details. Click here to see the algorithm types available for this system event.
Script Launching	This algorithm sets the script launching option for the script associated with a given state, if any. Refer to Script Launching Option for the details. Click here to see the algorithm types available for this system event.

Next Statuses

Use the **Next Statuses** grid to define the statuses a user can transition a case into while it's in this state. Refer to Valid States versus State Transition Rules for more information. Please note the following about this grid:

- Use **Action Label** to indicate the verbiage to display on the action button used to transition to this status.
- **Sequence** controls the order of the buttons that appear on Case - Main. Refer to Buttons Are Used To Transition A Case Into A State for more information.
- **Use as Default** controls which button (if any) is the default button.
- **Transition Condition** may be configured to identify a common transition path for cases of this type in the current state. This transition condition may then be referenced across multiple case types. You'll need to add values to Look Up table field **TR_COND_FLG** that fit the typical transitions for your case types (e.g. Ok, Error, etc.).

By assigning the transition condition value to a given "next status", you can design your Enter State transition or Auto-Transition logic to utilize those flag values *without specifying a status particular to a given case type*. Thus, similar logic may be used across a range of case types to transition a case into, for example, the next Ok state for the case's current status.

- **Transition Role** controls whether only the System or both System and User have the ability to transition a case into a given "next status".
- You can use the status description hyperlink to open the Status accordion to the respective status.
- When you initially set up a case type, none of the statuses will reside on the database and therefore you can't use the search to define a "next status". We recommend working as follows to facilitate the definition of this information:
    - Leave the Next Statuses grid blank when you initially define a case type's statuses
    - After all statuses have been saved on the database, update each status to define its Next Statuses (this way, you can use the search to select the status).

Required Characteristics

Use the **Required Characteristics** grid to define characteristics that are required when a case enters this state. Only Optional characteristics defined on the main page appear in this grid. Refer to Required Fields Before A Case Enters A State for more information.

Defining Umbrella Agreement Options

Your organization may use umbrella agreement functionality to manage many situations, including (but not limited to) the following:

- Creating a "negotiated contract" with a large customer with many sites.
- Creating a "negotiated contract" with all customers in a common geographic area, for example a specific city or an apartment complex
- Grouping and managing the proposal service agreements created during the quoting process
- Managing the renewal process for special contracts
- Applying override rate terms to one or more service agreements for a given effective period
- more...

> **NOTE:**
> **Separately module.** The umbrella agreement functionality is part of the Contract Management module. If this module is not applicable to your business you may turn it off. Refer to Turn Off A Function Module for more information.

The Big Picture Of Umbrella Agreements

The topics in this section provide background information about how to configure the system to support your umbrella agreement requirements.

Renewable Umbrella Agreements

When defining your umbrella agreement types, you must determine if umbrella agreements of this type are renewable. If so, you must set the renewal flag on the UA type to Renewable and determine the number of days prior to the end date of the umbrella agreement that the renewal process should start.

In addition, you must decide on your renewal procedure and design an appropriate renewal algorithm. You may use one of the sample renewal algorithms provided by the system.

- The sample algorithm URNW-TD creates a To Do entry. Using this algorithm you can direct the To Do entry to an appropriate role to determine if the umbrella agreement should be renewed. If the umbrella agreement is associated with an account management group, the To Do entry could be assigned to a role defined for that AMG.
- The sample algorithm URNW-CA creates a case. Using this algorithm you can create an appropriate case to help track the steps required for renewing an umbrella agreement.

If the base algorithms do not provide the logic your organization needs, you may create a new algorithm.

NOTE:
It is the responsibility of the renewal algorithm to change the renewal date as per the business requirements. If renewal requires manual review, the algorithm should reset the date to blank. If the algorithm can determine a new renewal date, the date should be changed to the appropriate value.

Terms Of Service Covered Entities

For each terms of service type you design, consider whether or not a TOS created for this type should reference one or more entities in the system that are "covered" by the terms of service. Following are some examples of covered entities:

- A special contract is created for all the McDonalds franchises in your service area. When creating the terms of service record for the umbrella agreement used for this contract, you want the user to link the person record representing the McDonalds corporate entity as a covered entity for the TOS.
- An umbrella agreement is created for a large customer with several types of services. Perhaps you want to group the services in the contract based on type of service. The service type may be set up as a covered entity.

The covered entities could also be used to denote eligibility. Refer to Linking To A UA For A Group Of Premises for an example of using the covered entity collection for eligibility.

The covered entities are actually characteristics. You must define the desired characteristic types if they do not already exist. In this case, they would be foreign key characteristic types. The characteristic types that you want to use must reference a characteristic entity of TOS covered entity.

A terms of service record may only reference covered entity types that you designate on the Terms of Service type record. When designing the covered entities that are allowed for TOS records of a given type, you may also configure the Terms of Service type to indicate that one or more covered entity types are required.

Refer to Terms Of Service Type - Main for more information.

Overriding Rate Terms

If your organization has business rules that require a service agreement's rate terms to be overridden, you can accomplish this using umbrella agreements and terms of service records. There are two levels of overriding that are possible:

- Override rate schedule
- Override other terms on the rate, including contract riders, contract terms and tax exemptions

To override any rate terms, the service agreement must be linked to a terms of service record that references a template service agreement. In addition, you must configure settings in the system to indicate whether or not overriding of terms is allowed.

- The SA type for the customer's service agreement indicates whether or not the rate schedule may be overridden using the rate source flag.
  - If the rate source value is Check TOS First, then SA, when applying the rate, the system determines if the SA is linked to a TOS effective at the time of the bill with a template SA. If so, the rate schedule on the template SA is used.
  - If the rate source value is Check SA Only, when applying the rate, the always uses the rate schedule on the customer's SA (regardless of whether or not the SA is linked to a TOS effective at the time of the bill with a template SA).
- If any bill factor on one of calculation rule indicates that it is eligible for tax exemptions, contract terms or contract riders, the terms of service usage flag on the bill factor indicates where the system should look for the appropriate value.

The flag tells the system to find applicable information either on the TOS's template SA only, on the customer's SA only or it should first check the TOS, then check the customer's SA.

The following table illustrates the different scenarios possible when configuring your bill factors and the resulting behavior. The term "standard behavior" in the tables below indicate that the standard behavior as described in Defining Bill Factors is followed.

	TOS Usage	Template SA	Customer SA	BF Char Source
Contract Rider Applicability is checked	Check TOS First, then SA	Found	Not checked	If source = SA, the template SA's characteristic collection is checked. If source <> SA, the standard behavior is followed.
		Not Found	Found	If source = SA, the customer SA's characteristic collection is checked. If source <> SA, the standard behavior is followed
		Not Found	Not Found	N/A
	Check TOS Only	Found	Not checked	If source = SA, the template SA's characteristic collection is checked. If source <> SA, the standard behavior is followed
		Not Found	Not checked	N/A
	Check SA Only	Not checked	Standard behavior	

	TOS Usage	Template SA	Customer SA	BF Value Source	BF Char Source
Value in Contract Term is checked	Check TOS First, then SA	Found	Not checked	Template SA	N/A
		Not Found	Found	Customer SA	N/A
		Not Found	Not Found	Bill Factor Value (based on Char Source)	If source = SA, the template SA's characteristic collection is checked. If source <> SA, the standard behavior is followed
	Check TOS Only	Found	Not checked	Template SA	N/A
		Not Found	Not checked	Bill Factor Value (based on Char Source)	If source = SA, the template SA's characteristic collection is checked. If source <> SA, the standard behavior is followed

	Check SA Only	Not checked	Standard behavior		

	TOS Usage	Template SA	Customer SA	BF Value Source
Tax Exemption is checked	Check TOS First, then SA	Found	Not checked	Template SA
		Not Found	Found	Customer SA
		Not Found	Not Found	No value
	Check TOS Only	Found	Not checked	Template SA
		Not Found	Not checked	No value
	Check SA Only	Not checked	Standard behavior	

Note that proration logic is followed for the above functionality as expected. For example, imagine that a bill factor with Value in Contract Terms is proratable and the TOS usage value is Check TOS First, then SA. If the template SA has a contract value that is only in effect for the first 10 days of a bill, that value is used for the first 10 days and the remaining days in the bill take the value from the SA (if applicable) or the bill factor value as per the table rules above.

NOTE:
When designing your terms of service type, you must indicate whether a template service agreement is optional, required or not allowed. When the template SA is required, the user setting up the TOS record for this TOS type must link the correct service agreement as the template SA. If a new template service agreement is required, the new SA must be created first.

Setting Up Umbrella Agreement Options

The topics in this section describe how to set up umbrella agreement options.

Configure SA Types for Umbrella Agreements

For each type of service agreement that may be linked to a terms of service record, you must configure the Rate Source for the SA type to indicate if the rate should always be taken from the service agreement (Check SA Only) or if the rate should be taken from the terms of service record for the service agreement, if any (Check TOS First, then SA).

FASTPATH:
Refer to Overriding Rate Terms for more information.

NOTE:
The rate source is only visible on SA type if the Contract Management module is not turned off.

Configure Bill Factors for Umbrella Agreements

For each bill factor that has a value in contract term, contract rider applicability or tax exemption applicability, you must indicate where the system should look for a value when a service agreement is linked to a terms of service with a template SA and where this bill factor is referenced by the rate being applied. Using the terms of service usage flag, tell the system to Check TOS Only, Check SA Only or Check TOS First, then SA.

●

FASTPATH:
Refer to Overriding Rate Terms for more information.

➤

NOTE:
The terms of service usage flag is only visible on bill factor if the Contract Management module is not turned off.

Configure A Campaign To Link An SA To A Terms Of Service

The system provides a pair of column reference algorithm types that support linking a new service agreement to an existing terms of service record via an order. The pair includes a validation algorithm type CRVL-SATOS and a posting algorithm type CRPS-SATOS.

In order to take advantage of this functionality, you must:

- Create an algorithm for each of the above algorithm types. Only one algorithm is needed for each algorithm type because there are no parameters for either one.
- Setup a unique column reference for the terms of service ID column reference code and plug in the new validation and posting algorithms on the record.
- For campaigns whose orders link a new service agreement to an existing umbrella agreement, make sure to define a miscellaneous field for the terms of service ID. Note, the miscellaneous field must reference the column reference that you setup above.

Based on your business practice, you may want to further configure your campaign / package for linking a service agreement to an existing terms of service. Refer to Linking A Service Agreement To A TOS Through Orders for more information.

Setting Up Umbrella Agreement Types

The umbrella agreement type transaction is used to maintain your UA types. The topics in this section describe how to use this transaction.

Umbrella Agreement Type - Main

Open **Admin** > **Umbrella Agreement** > **Umbrella Agreement Type** > **Add** to define basic information about an umbrella agreement type.

Description of Page

Enter a unique **Umbrella Agreement Type** code and **Description** for the UA type.

Use **Renewal** to indicate whether an umbrella agreement of this type is Renewable or Not Renewable. For renewable UA types, indicate the **Renewal Days Before Expiration**. The renewal date for umbrella agreements of this type are calculated based on this number of days before the end date.

Use the characteristics collection to define characteristics that can be defined for umbrella agreements of a given type. Use **Sequence** to control the order in which characteristics are defaulted. Turn on the **Required** switch if the **Characteristic Type** must be defined on umbrella agreements of a given type. Turn on the **Default** switch to default the **Characteristic Type** when umbrella agreements of the given type are created. Enter a **Characteristic Value** to use as the default for a given **Characteristic Type** when the **Default** switch is turned on.

> **NOTE:**
> **Characteristic Types.** You can only choose characteristic types defined as permissible on an umbrella agreement record. Refer to *Setting Up Characteristic Types & Their Values* for more information.

Enter the valid **Terms of Service Type**s that may be referenced for umbrella agreements of this type.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference CI_UA_TYPE.

Umbrella Agreement Type - Algorithms

Open **Admin** > **Umbrella Agreement** > **Umbrella Agreement Type** > **Search** and navigate to the **Algorithms** page.

Description of Page

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 (descriptions of all possible events are provided below).
- 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.

The following table describes each **System Event** for which you can define algorithms.

System Event	Description
UA Renewal	Algorithms of this type are executed by the UARENEW background process when an umbrella agreement has reached its renewal date. Only one renewal algorithm is allowed for a given UA type. Click here to see the algorithm types available for this system event.

Setting Up Terms Of Service Types

The terms of service type transaction is used to maintain your Terms of Service types. The topics in this section describe how to use this transaction.

Terms Of Service Type - Main

Open **Admin** > **Umbrella Agreement** > **Terms of Service Type** > **Add** to define basic information about terms of service types.

Description of Page

Enter a unique **Terms of Service Type** code and **Description** for the TOS type.

Use **Template SA Usage** to indicate whether a template SA is Optional, Required or Not Allowed. Refer to Overriding Rate Terms for more information about using template SAs.

Use the characteristics collection to define characteristics that can be defined for terms of service records of a given type. Use **Sequence** to control the order in which characteristics are defaulted. Turn on the **Required** switch if the **Characteristic Type** must be defined on terms of service records of a given type. Turn on the **Default** switch to default the **Characteristic Type** when terms of service records of the given type are created. Enter a **Characteristic Value** to use as the default for a given **Characteristic Type** when the **Default** switch is turned on.

➤ **NOTE:**
Characteristic Types. You can only choose characteristic types defined as permissible on a terms of service record. Refer to *Setting Up Characteristic Types & Their Values* for more information.

Use the **Covered Entities** collection to define the types of entities that are "covered" by terms of service records of this type. Refer to Terms Of Service Covered Entities for more information

➤ **NOTE:**
Characteristic Types. You can only choose characteristic types defined as permissible for TOS covered entity. Refer to *Setting Up Characteristic Types & Their Values* for more information.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference CI_TOS_TYPE.

Terms Of Service Type - UA Types

Open **Admin** > **Umbrella Agreement** > **Terms of Service Type** > **Add** and navigate to the **UA Types** page.

Description of Page

Enter the valid **Umbrella Agreement Type**s that may reference this terms of service type.

Setting Up Terms of Service Cancel Reasons

Whenever a terms of service record is canceled, a cancel reason must be specified. Open **Admin** > **Terms of Service Cancel Reason** to specify such cancellation reasons.

Description of Page

Enter an easily recognizable **Terms of Service Cancel Reason** and **Description** for the terms of service cancellation reason.

Umbrella Agreements Configuration Examples

This section provides examples of how you would configure the system to support various business scenarios.

Linking A Service Agreement To A TOS Through Orders

As described in Configure A Campaign To Link An SA To A Terms Of Service, the system has provided a pair of column reference algorithms to enable your users to link a new service agreement to an existing terms of service record (and its umbrella agreement) when starting service via the order transaction.

When designing your campaign and packages to use this option, consider the scenarios where you may use this to determine whether or not you require additional configuration. The following topics describe some possible scenarios.

Linking To A UA For A Given Account

Imagine that you would use this logic when an important customer is starting additional service and has an existing umbrella agreement for other services that the new service should be added to. The user creating the order is probably the account manager for that account and would know the correct terms of service record to link the new service to.

For this scenario, you do not need to provide any "help" to the user to indicate that linking to a TOS is an option or to help select the correct TOS. You simply need to ensure that the campaign that would be used by this account manager to start service for the important customer simply includes a question / miscellaneous field to link the SA to the terms of service.

Linking To A UA For A Group Of Premises

Imagine you have created an umbrella agreement for a group of premises that are not actively managed by a specific user. For example, perhaps there are several apartment complexes that have negotiated a special rate with your company. When new customers sign up for service at a premise that belongs to one of these apartments, the user should be informed that there is a special contract for this premise. The user may also need help selecting the correct terms of service record. To accomplish this logic, you must define additional algorithms to help your users.

For example, perhaps your standard campaign for residential electric customers includes a package to be used by premises linked to one of these apartment buildings and should only be eligible to one of those premises. You must create an eligibility algorithm for this package. Its logic could be designed based on the following assumptions:

- The qualifying premises are linked to a parent premise
- The parent premise is linked to the appropriate terms of service record in its covered entity collection

The eligibility algorithm checks to see if the order's premise is linked to a parent premise that is referenced on a terms of service record for an umbrella agreement that is currently in effect. It could also be restricted to search for a specific umbrella agreement type or terms of service type if appropriate.

NOTE:
The campaign for this scenario must include a question / miscellaneous field to link the SA to the terms of service. It should be configured with an applicability of only applicable on package. The package that is used to include the SA into the terms of service for the apartment complex would include this question / miscellaneous field.

Once the user chooses this package, user is prompted to provide the correct terms of service record via the question / miscellaneous field. To help the user, you should create a column reference retrieval algorithm to default the correct terms of service record for this premise. It would use the same logic as the eligibility algorithm described above. In other words, it would find the parent premise for this premise and then find the terms of service record covered entity collection includes this parent premise and whose umbrella agreement is currently in effect.

Reports Addendum

This chapter is an addendum to the general Defining and Designing Reports chapter.

Configuring the System to Invoke Oracle BI Publisher Enterprise Real-Time

In order to invoke Oracle BI Publisher Enterprise from within the system, it must be configured. This topic describes the steps at a high level that are required to configure your system to enable BI Publisher..

Specify Reporting Tool Options

The following information should be defined on Reporting Tool Options table:

- **Reporting Server** is the web server URL where the reporting tool is installed.
- **Reporting Folder** defines the name of the directory on the Oracle BI Publisher server where reports are located. The system's reporting tool algorithm constructs the URL to open Oracle BI Publisher Enterprise based on the information provided on Reporting Options. The same folder name should be used as a report folder in Oracle BI Publisher Enterprise where reports are published.

Reporting Tool Algorithm

The base product provides an installation algorithm plug-in spot called Reporting Tool. This plug- in spot should contain an algorithm that invokes the reporting tool real-time. The corresponding algorithm type, called algorithm reference: F1-BIPR-INV, invokes Oracle BI Publisher Enterprise.

This algorithm relies on information defined in the Reporting Options table: the reporting server and folder names for accessing the reporting tool. The algorithm constructs several strings, for example, the URL of the BI Publisher, information about the report being requested and user info in the format expected by Oracle BI Publisher Enterprise. The Java program returns the correct URL to the browser. The browser then opens a BI Publisher Enterprise window with the appropriate URL.

To use the algorithm reference: F1-BIPR-INV algorithm type to invoke Oracle BI Publisher Enterprise, perform the following steps:

1. Create an algorithm for this algorithm type.
2. In 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.

Interface with Oracle BI Publisher Enterprise Batch Scheduler

Presently the base product does not provide support for batch jobs with Oracle BI Publisher. Please use Oracle BI Publisher directly in order to schedule reports to run in batch.

Configuring the System for Batch Bill Print

This section describes the steps for configuring the applications to print bills in batch.

Configure Oracle BI Publisher for Batch Bill Print

1. Configure the Oracle BI Publisher Scheduler to allow report job submission.

   > **NOTE:**
   > Refer to the Oracle Fusion Middleware Administrator's Guide for Oracle Business Intelligence Publisher for more information.

2. Ensure that the CI_BILLPR report artifacts (CI_BILLPR.rtf, CI_BILLPR.xdo, CI_BILLPR.xdm) are deployed to corresponding BI Publisher catalog folder:

Configure Oracle Utilities Customer Care and Billing for Batch Bill Print

1. Go to **Admin** > **Reporting** > **Reporting Options** and configure the following values:
   - **Reporting Server from Browser** should point to the correct BI Publisher Server URL; for example, `http://<hostname>:port`
   - **Reporting Folder** is the name as the Catalog Folder in BI Publisher where CI_BILLPR is located; for example, CCB
   - **Reporting Engine User ID** is the BI Publisher User ID.
   - **Reporting Engine Password** is the BI Publisher Password. is the BI Publisher Password.
2. Create a custom Bill Route Type Extract Algorithm using the C1-BLEX-BIPAlgorithm Type. Ensure that the following algorithm soft parameters are set:
   - **Output Directory** is the directory in BI Publisher server where the generated output files will be placed, for example. `/tmp/`
   - **Report Absolute Path** is the path where CI_BILLPR.xdo is located in the BI Publisher Catalog, for example, `/CCB/CI_BILLPR/CI_BILLPR.xdo`
   - **Report Format** is the format of the generated output files; for example, `pdf`

- **Report Code** is the CI_BILLPR report code; for example, CI_BILLPR

3. **Go to Admin Menu, B, Bill Route Type** and configure the corresponding Bill Route Type (such as, POSTAL) to use the RTTYPOST Batch Control and the custom Bill Route Type Extract Algorithm.

Description of Sample Reports

This section provides an overview of each sample report supplied with Oracle Utilities Customer Care and Billing that may be found in the demonstration database. They may be used by your organization as they are or as a starting point for creating a new report.

> **NOTE:**
> **Account Security**. The sample reports provided with the product do NOT incorporate account security. If a user has been given security to view the report, then all the data in the report is available for viewing.

Active Severance Processes - CI_ACSVPR

Parameters

Parameter	Parameter Code	Description
Start Date	P_FROM_DT	Start date to use for reporting severance processes. If not defined, start date is set to 1900-01-01.
End Date	P_TO_DT	End date to use for reporting active severance processes. If not defined to date is set to the current date.

Report Description

This report is used to aid in the monitoring of active severance processes. The report details are used to review how the company is doing as far as the collection of outstanding debt, as well as to monitor the progress of currently active severance processes.

The report selects active severance processes whose creation date is between the input start and end dates.

Bill Print in BI Publisher - CI_BILLPR

Parameters

Parameter	Parameter Code	Description
CC&B user	P_USER_ID	No default value. Required. Returns an error message if user ID is null, empty or does not exist in database.

Batch Switch	P_BATCH_SW	For a single bill print, Batch Switch is 'No' and for multiple bill print, Batch Switch is 'Yes'.
First Bill	**P_FROM_BILL_ID**	First Bill Id to print for the Bill Range.
Last Bill	**P_TO_BILL_ID**	Last Bill Id to print for the Bill Range.
Batch Code	**P_BATCH_CD**	Passed in by the system the report is submitted in batch. Only bills with a Bill Routing that references this batch code are selected.
Batch Number	**P_BATCH_NBR**	Passed in by the system the report is submitted in batch. Only bills with a Bill Routing that references this batch number are selected.
Extract Algorithm	**P_EXTRACT_ALG**	Passed in by the system the report is submitted in batch. Only bills with a Bill Routing whose bill route type references this extract algorithm are selected.

Report Description

This report is used with BI Publisher to display all the information that appears on the customer's printed bill for the bill ID entered in the input parameters. Refer to the description of the parameters above. The bill is only displayed if the bill is complete.

See below for details of the information extracted for the report. In addition, the sample report layout provided for the base product includes:

- Payment coupon.
- Form for change of address.
- A graph of consumption for current charges.

The following sections highlight the data that is extracted for this report.

Bill Information

Field Name	Format	Source/Value/Description
SORT_KEY	A12	Primary sort for the report - currently sorted by Bill Id
BILL_ID	A12	CI_BILL
SEQNO	N5	CI_BILL_ROUTING
PER_ID	A10	CI_BILL_ROUTING
NBR_BILL_COPIES	N1	CI_BILL_ROUTING
ENTITY_NAME1	A64	CI_BILL_ROUTING. Mailing Entity Name 1.
ENTITY_NAME2	A64	CI_BILL_ROUTING. Mailing Entity Name 2.
ENTITY_NAME3	A64	CI_BILL_ROUTING. Mailing Entity Name 3.
COUNTRY	A3	CI_BILL_ROUTING
ADDRESS1	A254	CI_BILL_ROUTING
ADDRESS2	A254	CI_BILL_ROUTING

ADDRESS3	A254	CI_BILL_ROUTING
ADDRESS4	A254	CI_BILL_ROUTING
CITY	A30	CI_BILL_ROUTING
COUNTY	A30	CI_BILL_ROUTING
STATE	A6	CI_BILL_ROUTING
POSTAL	A12	CI_BILL_ROUTING
IN_CITY_LIMIT	A1	CI_BILL_ROUTING
BATCH_CD	A8	CI_BILL_ROUTING
BATCH_NBR	N10	CI_BILL_ROUTING
NO_BATCH_PRT_SW	A1	CI_BILL_ROUTING
ACCT_ID	A10	CI_BILL
BILL_STAT_FLG	A2	CI_BILL
BILL_DT	Date	CI_BILL
DUE_DT	Date	CI_BILL
ACCT_ENTITY_NAME	A64	Get Main Person from CI_PER_NAME. Account Entity Name
LANGUAGE_CD	A3	Person's Language from CI_PER
CURRENCY_CD	A3	CI_ACCT
CUR_SYMBOL	A4	CI_CURRENCY_CD
DECIMAL_POSITIONS	N1	CI_CURRENCY_CD
CUR_POS_FLG	A2	CI_CURRENCY_CD
OPEN_ITEM_SW	A1	CI_CUST_CL
BILL_COPY	A1	Bill copy number

Bill Segment Information

Field Name	Format	Source/Value/Description
BSEG_ID	A12	CI_BSEG
SA_ID	A10	CI_BSEG
START_DT	Date	CI_BSEG
END_DT	Date	CI_BSEG
BILL_CYC_CD	A4	CI_BSEG
WIN_START_DT	Date	CI_BSEG
EST_SW	A1	CI_BSEG
CLOSING_BSEG_SW	A1	CI_BSEG
PREM_ID	A10	CI_BSEG
CIS_DIVISION	A5	CI_SA
SA_TYPE_CD	A8	CI_SA

SAT_DESCR	A60	CI_SA_TYPE_L
BILL_PRT_PRIO_FLG	A2	CI_SA_TYPE
GRAPH_UOM_CD	A4	CI_SA_TYPE
HEADER_SEQ	N3	CI_BSEG_CALC
BCALC_START_DT	Date	CI_BSEG_CALC
BCALC_END_DT	Date	CI_BSEG_CALC
RS_CD	A8	CI_BSEG_CALC
EFFDT	Date	CI_BSEG_CALC
BILLABLE_CHG_ID	A12	CI_BSEG_CALC
CALC_AMT	N15.2	CI_BSEG_CALC
BCALC_DESC_BILL	A80	CI_BSEG_CALC
SEQNO	N5	CI_BSEG_CALC_LN
PRT_SW	A1	CI_BSEG_CALC_LN
APP_IN_SUMM_SW	A1	CI_BSEG_CALC_LN
CALC_LN_AMT	N15.2	CI_BSEG_CALC_LN
DST_ID	A10	CI_BSEG_CALC_LN
MSR_PEAK_QTY_SW	A1	CI_BSEG_CALC_LN
DESCR_LN	A80	CI_BSEG_CALC_LN
STATE	A6	CI_PREM
CITY	A30	CI_PREM
POSTAL	A12	CI_PREM
SIBLING_ID	A12	CI_FT

Account Summary Information

Field Name	Format	Source/Value/Description
CURRENCY_CD	A3	Derived from CI_BILL_SA or CI_FT.
SUM_CUR_AMT	S15.2	Get Bill's Current Balance from CI_BILL_SA Bill's Current Charge from CI_FT Bill's Current Correction Charge from CI_FT Bill's Current Adjustment from CI_FT Bill's Current Payment from CI_FT
SUM_TOT_AMT	S15.2	Get Bill's Total Balance from CI_BILL_SA Bill's Total Charge from CI_FT Bill's Total Correction Charge from CI_FT Bill's Total Adjustment from CI_FT Bill's Total Payment from CI_FT
BILL_PRT_FLG	A4	Set as follows:

		ASBL - Indicator for Bill's Balance from CI_BILL_SA ASBS - Indicator for Bill's Current Charge from CI_FT ASBC - Indicator for Bill's Correction from CI_FT ASAD - Indicator for Bill's Adjustment from CI_FT ASPS - Indicator for Bill's Payment from CI_FT

Current Charge Information

Field Name	Format	Source/Value/Description
SUM_CUR_AMT	S15.2	Derived from CI_FT (Only the current charge bill segments.)
SUM_TOT_AMT	S15.2	Derived from CI_FT (Only the current charge bill segments.)
CURRENCY_CD	A3	CI_BILL_SA or CI_FT
DEBT_CL_CD	A4	Debt Class of SA Type
DESCR	A60	Debt Class Description

Premise Information

One record for every premise linked to the bill.

Field Name	Format	Source/Value/Description
PREM_ID	A10	Premise Id obtained from Bill Segment
PREM_TYPE_CD	A8	CI_PREM
LL_ID	A10	CI_PREM
MAIL_ADDR_SW	A1	CI_PREM
TREND_AREA_CD,	A8	CI_PREM
COUNTRY	A3	CI_PREM
ADDRESS1	A254	CI_PREM
ADDRESS2	A254	CI_PREM
ADDRESS3	A254	CI_PREM
ADDRESS4	A254	CI_PREM
CITY	A30	CI_PREM
NUM1	A6	CI_PREM

NUM2	A4	CI_PREM
HOUSE_TYPE	A2	CI_PREM
COUNTY	A30	CI_PREM
STATE	A6	CI_PREM
POSTAL	A12	CI_PREM
GEO_CODE	A11	CI_PREM
IN_CITY_LIMIT	A1	CI_PREM

Auto Pay Information

Field Name	Format	Source/Value/Description
BILL_ID	A12	CI_APAY_CLR_STG
APAY_SRC_CD	A12	CI_APAY_CLR_STG
EXT_ACCT_ID	A50	CI_APAY_CLR_STG
SCHED_EXTRACT_DT	Date	CI_APAY_CLR_STG. The date that the automatic payment will be downloaded.
APAY_SRC_NAME	A30	CI_APAY_SRC_L
APAY_SRC_DESCR	A60	CI_APAY_SRC_L
TNDR_TYPE_DESCR	A60	CI_TENDER_TYPE_L
TENDER_TYPE_CD	A4	CI_TENDER_TYPE

Bill Segment Readings

Field Name	Format	Source/Value/Description
BSEG_ID	A12	CI_BSEG_READ
SP_ID	A10	CI_BSEG_READ
SEQNO	N5	CI_BSEG_READ
REG_CONST	S12.6	CI_BSEG_READ
USAGE_FLG	A2	CI_BSEG_READ
USE_PCT	S3	CI_BSEG_READ
HOW_TO_USE_FLG	A2	CI_BSEG_READ
MSR_PEAK_QTY_SW	A1	CI_BSEG_READ
UOM_CD	A30	CI_BSEG_READ
TOU_CD	A30	CI_BSEG_READ
SQI_CD	A30	CI_BSEG_READ
START_REG_READ_ID	A12	CI_BSEG_READ

START_READ_DTTM	DTTM	CI_BSEG_READ
START_REG_READING	S15.6	CI_BSEG_READ
END_REG_READ_ID	A12	CI_BSEG_READ
END_READ_DTTM	DTTM	CI_BSEG_READ
END_REG_READING	S15.6	CI_BSEG_READ
MSR_QTY	S18.6	CI_BSEG_READ
FINAL_UOM_CD	A4	CI_BSEG_READ
FINAL_TOU_CD	A8	CI_BSEG_READ
FINAL_SQI_CD	A30	CI_BSEG_READ
FINAL_REG_QTY	S18.6	CI_BSEG_READ
BADGE_NBR	A30	CI_MTR
NBR_OF_DGTS_RGT	S2	CI_REG

Bill Segment Consumption History

Field Name	Format	Source/Value/Description
SA_ID	A10	CI_BSEG
START_DT	Date	CI_BSEG
BSEG_ID	A12	CI_BSEG
END_DT	Date	CI_BSEG
UOM_CD	A30	CI_BSEG_SQ
TOU_CD	A30	CI_BSEG_SQ
SQI_CD	A30	CI_BSEG_SQ
QTY	S18.6	Service Quantity from CI_BSEG_SQ

Payment Information

One record for every payment linked to the bill.

Field Name	Format	Source/Value/Description
PAY_DT	Date	CI_PAY_EVENT
FT_TYPE_FLG	A2	Pay or Pay Cancel ('PS' or 'PX')
CAN_RSN_CD	A4	CI_PAY
CURRENCY_CD	A3	CI_FT.
CUR_AMT	S15.2	Derived from CI_FT. Sum of CUR_AMT for the bill with FT Type Flag = (PS or PX) and Show on Bill Switch = 'Yes'

TOT_AMT	S15.2	Derived from CI_FT. Sum of CUR_AMT for the bill with FT Type Flag = (PS or PX) and Show on Bill Switch = 'Yes'
TOT_PREV_BAL	S15.2	Calculated as Previous Balance = Ending Balance - (current charges + payments + adjustments + corrections)
PAY_CAN_RSN_DESCR	A60	Description of cancel reason code. (Note, that this is retrieved in a special stored procedure that retrieves all the cancel reason codes and descriptions.)

Adjustment Information

One record for every adjustment linked to the bill.

Field Name	Format	Source/Value/Description
BILL_ID	A12	CI_FT
ACCOUNTING_DT	Date	CI_FT
ARS_DT	Date	CI_FT
PARENT_ID	A12	CI_FT
SIBLING_ID	A12	CI_FT
FT_ID	A12	CI_FT
CUR_AMT	S15.2	CI_FT
TOT_AMT	S15.2	CI_FT
FT_TYPE_FLG	A2	Adjustment or Adjustment Cancel ('AD' or 'AX')
SHOW_ON_BILL_SW	A1	CI_FT
CURRENCY_CD	A3	CI_FT
XFERRED_OUT_SW	A1	CI_FT
ADJ_TYPE_CD	A8	CI_ADJ
CAN_RSN_CD	A4	CI_ADJ
SA_ID	A10	CI_ADJ
CHAR_PREM_ID	A10	CI_SA
DESCR_ON_BILL	A254	CI_ADJ_TYPE_L
ADJ_CAN_RSN_DESCR	A60	Description of cancel reason code. (Note that this is retrieved in a special stored procedure that retrieves all the cancel reason codes and descriptions.)

Bill Message Information

One record for every bill message linked to the bill.

Field Name	Format	Source/Value/Description
MSG_PRIORITY_FLG	A12	CI_BILL_SA or CI_FT
INSERT_CD	S15.2	CI_BILL_SA or CI_FT
MSG_ON_BILL	S15.2	CI_BILL_SA or CI_FT

Billed Revenues by Rate - CI_BILREV

Parameters

Parameter	Parameter Code	Description
Accounting Period	P_ACCT_PERIOD	Defines the accounting period used for the report. A valid fiscal year and accounting period for a valid accounting calendar must be provided.
Account Type Characteristic	P_CHAR_TYPE	Defines a Characteristic Type for a characteristic linked to the Distribution Code to define an Account Type.
Account Type	P_REV_ACCTY_CHAR	Account Type Char Value for Revenue related GL Accounts. The char type defined for this parameter should match the Char Type code defined as parameter #2. The parameter value indicated for this parameter should be one that represents revenue accounts.

Report Description

This is an analysis report for the billed revenues for an accounting period according to the various rates, which were in effect in the system. The information in this report helps to adjust rates in order to achieve better financial results and comply with regulations and market trends.

This report selects all records in the financial transaction GL collection that satisfy the following criteria:

- The financial transaction is frozen.
- The Accounting Date on the financial transaction within input accounting period (parameter 1)
- The distribution code associated with the GL entry has a characteristic type and value that matches the input Account Type Characteristic and Account Type (parameters 2 & 3)

> **NOTE:**
> **Performance Consideration.** If your implementation chooses to use this report, you may consider adding an index to the CI_FT table on ACCOUNTING_DT to aid in performance. When making this decision, carefully weigh the benefit of improving report performance against the possible degradation to the performance of day-to-day processing as a result of defining a new index. In addition, many companies opt

> to create a reporting database that is a shadow of production to ensure that indexes defined to benefit reports may be created without any affect on the production environment.

Case Statistics By Case Type - CI_CSESTS

Parameters

Parameter	Parameter Code	Description
Start Date	P_FROM_DT	See the report's description for how this field is used. If start date is not specified, it is defaulted to 7 days prior to the end date.
End Date	P_TO_DT	See the report's description for how this field is used. If end date is not specified, it is defaulted to the current processing date.
Case Condition (Open, Closed)	P_COND_FLG	If specified, only cases in this condition are included in the report. If left blank, the reports produces statistics for both open and closed cases.

Report Description

This report provides two types of statistics:

1. Open cases whose creation date falls between the input Start Date and End Date (inclusive)

2. Closed cases whose closing date falls between the input Start Date and End Date (inclusive)

The third parameter is only used if you want to restrict the statistics to only open or closed cases. If you leave this parameter blank, both open and closed statistics will be produced.

The following information is provided in graphical format:

- Number of cases by case type
- Percentage of cases by case type

Case Statistics for a Given Status - CI_CSESGS

Parameters

Parameter	Parameter Code	Description
Start Date	P_FROM_DT	See the report's description for how this field is used. If start date is not specified, it is defaulted to 7 days prior to the end date.
End Date	P_TO_DT	See the report's description for how this field is used.

		If end date is not specified, it is defaulted to the current processing date.
Case Type/Status	P_CASE_STATUS_CD	This is the desired Case Type and Status that will be reported on.
Responsible User	P_CASE_OWNER	If specified, only cases with this responsible user are included in the report.
First Bucket High Limit	P_B1_LIMIT	Cases that took <= this number of days to reach the given status will be grouped together for statistical reporting.
Second Bucket High Limit	P_B2_LIMIT	Cases that took <= this number of days but more than the first bucket high limit to reach the given status will be grouped together for statistics reporting.
Third Bucket High Limit	P_B3_LIMIT	Cases that took <= this number of days but more than the second bucket high limit to reach the given status will be grouped together for statistics reporting. Cases that took more than this number of days are included in another group.

Report Description

This report shows cases of a given case type that transitioned to a given status during a given date range.

Graphs are printed to show the number and percentage of cases grouped by the time it took to reach the status. These statistics are grouped into age buckets whose size is controlled by the last 3 parameters.

Summary statistics are also printed showing the minimum, maximum, average and median times for these cases.

Collection Summary - CI_CLLSUM

Parameters

Parameter	Parameter Code	Description
Start Date	P_FROM_DT	If from date is not set, the default is one month prior to the current date.
End Date	P_TO_DT	End date of the date range. If not defined by user, it is set to one month after the current date

Report Description

This report provides detailed monthly summary information of all collection activities. The report is typically used by a collection department for resource planning and performance review purposes.

This report selects pending and completed collection events whose event trigger date falls between the input start and end dates.

Customer Contacts by Type - CI_CUSTCN

Parameters

Parameter	Parameter Code	Description
Start Date	P_FROM_DT	Start date to use for reporting customer contacts. If not defined, the start date is set to the current date minus 7 days.
End Date	P_TO_DT	End date to use for reporting customer contacts. If not defined, End Date is set to the current date.
Customer Contact Class / Type	P_CC_TYPE_CD	Specify a Customer Contact Class / Type to restrict the report output to a specific class / type.

Report Description

This report lists all customer contacts in the system created within the input date range. You may optionally restrict the report to customer contacts for a given Customer Contact Class / Type (parameter 3).

> **NOTE:**
> **Graphs.** The information on this report is shown in both textual and graphical formats.
>
> **Performance Consideration.** If your implementation chooses to use this report, you may consider adding an index to the CI_CC table on CC_DTTM to aid in performance. When making this decision, carefully weigh the benefit of improving report performance against the possible degradation to the performance of day-to-day processing as a result of defining a new index. In addition, many companies opt to create a reporting database that is a shadow of production to ensure that indexes defined to benefit reports may be created without any affect on the production environment.

Customers with Life-Support / Sensitive-Load - CI_PMLSSL

Parameters

Parameter	Parameter Code	Description
Service Type	P_SERVICE_TYPE	Service type used to restrict the report to premises with services of this type.

Report Description

This reports displays a detailed list of premises that are coded with Life-Support (LS) or Sensitive-Load (SL) information. It may optionally restrict the output to premises with service points for input service type.

This information is used by a company to make sure that customers at these premises are dealt with appropriately in the case of an outage (planned and unplanned) or service cut due to non-payment.

The report provides detailed information about the premise and the related facility elements providing service to that premise (e.g. substation, feeder and node for electric service).

> **NOTE:**

If ANY person connected to an account has an LS/SL indication in the Person record, then all the premises connected to this account will be treated as LS/SL. This means, for example, that a premise connected to an account that has a 3rd party guarantor with LS/SL is treated as a premise with LS/SL.

Field Order Print in BI Publisher- CI_FOPRNT

NOTE:
Field Order Print in batch for BI Publisher is not supported yet.

Parameters

Parameter	Parameter Code	Description
CC&B user	P_USER_ID	No default value. Required. Returns an error message if user ID is null, empty or does not exist in database.
Batch Switch	P_BATCH_SW	For a single FO print, Batch Switch is 'No' and for multiple FO print, Batch Switch is 'Yes'.
First FO	**P_FROM_FO_ID**	First FO ID to print for the FO Range.
Last FO	**P_TO_FO_ID**	Last FO ID to print for the FO Range.
Batch Code	**P_BATCH_CD**	Batch Code passed in by the system the report is submitted in batch. Only field orders that reference this batch code are selected.
Batch Number	**P_BATCH_NBR**	Batch Code passed in by the system the report is submitted in batch. Only field orders that reference this batch number are selected.
Extract Algorithm	**P_EXTRACT_ALG**	Passed in by the system the report is submitted in batch. Only field orders whose dispatch groups reference this FO extract algorithm are selected.

Report Description

This report is used with BI Publisher to display the Field Order information for the field order ID entered in the input parameters. Refer to the description of the parameters above. The field order will only be displayed if the field order's status is dispatched.

The following sections highlight the data that is extracted for this report.

Field Order Information

Field Name	Format	Source/Value/Description
LANGUAGE_CD	A3	SC_USER

SORT_KEY	A10	FO Id
FO_ID	A10	CI_FO
PREM_ID	A10	CI_FO
FO_SCHED_DTTM	Date	CI_FO.SCHED_DTTM
FO_STATUS_FLG	A2	CI_FO
WORK_DTTM	Date	CI_FO
DISP_GRP_CD	A8	CI_FO
REP_CD	A8	CI_FO
WORKED_BY	A8	CI_FO
EXTRACT_NEXT_SW	A1	CI_FO
EXTRACT_DTTM	Date	CI_FO
FO_DESCR	A254	CI_FO.DESCR254
BATCH_CD	A8	CI_FO_STG_DWN
BATCH_NBR	N10	CI_FO_STG_DWN
FA_ID	A10	CI_FA
SP_ID	A10	CI_FA
FA_TYPE_CD	A8	CI_FA
FA_PRIORITY_FLG	A2	CI_FA
FA_CREATED_BY_FLG	A2	CI_FA
FA_SCHED_DTTM	Date	CI_FA.SCHED_DTTM
FA_STATUS_FLG	A2	CI_FA
ELIG_DISPATCH_SW	A1	CI_FA
CRE_DTTM	Date	CI_FA
INSTRUCTIONS	A254	CI_FA
TEST_SEL_ID	A10	CI_FA
FA_DESCR	A254	CI_FA.DESCR254
FA_CAN_RSN_CD	A8	CI_FA_STEP
STEP_SEQ_NBR	N3	CI_FA_STEP
FA_STEP_TY_ACT_FLG	A2	CI_FA_STEP
STP_ENTITY_FLG	A4	CI_FA_STEP
SP_MTR_HIST_ID	A10	CI_FA_STEP
SP_ITEM_HIST_ID	A10	CI_FA_STEP
MR_ID	A12	CI_FA_STEP
MTR_CFG_MTR_ID	A10	CI_FA_STEP
MTR_ID	A10	CI_FA_STEP
ITEM_ID	A10	CI_FA_STEP
CC_ID	A10	CI_FA_STEP
SPAWNED_FA_ID	A10	CI_FA_STEP
ACCT_ID	A10	CI_FA_STEP

DV_TEST_ID	A10	CI_FA_STEP
PREM_TYPE_CD	A8	CI_PREM
LL_ID	A10	CI_PREM
MAIL_ADDR_SW	A1	CI_PREM
KEY_SW	A1	CI_PREM
KEY_ID	A10	CI_PREM
OK_TO_ENTER_SW	A1	CI_PREM
MR_INSTR_CD	A4	CI_PREM
MR_INSTR_DETAILS	A250	CI_PREM
MR_WARN_CD	A4	CI_PREM
TREND_AREA_CD	A8	CI_PREM
COUNTRY	A3	CI_PREM
ADDRESS1	A254	CI_PREM
ADDRESS2	A254	CI_PREM
ADDRESS3	A254	CI_PREM
ADDRESS4	A254	CI_PREM
CITY	A30	CI_PREM
NUM1	A6	CI_PREM
NUM2	A5	CI_PREM
HOUSE_TYPE	A2	CI_PREM
COUNTY	A30	CI_PREM
STATE	A6	CI_PREM
POSTAL	A12	CI_PREM
GEO_CODE	A11	CI_PREM
IN_CITY_LIMIT	A1	CI_PREM
FA_STEP_TY_DESCR	A60	CI_LOOKUP.DESCR / FA Step Type Action Flag
OPTIONAL_SW	A1	CI_FA_STEP_TYPE
FA_STEP_TYPE_DESCR	A60	CI_FA_STEP_TYPE_L.DESCR
REP_DESCR	A60	CI_REP_L.DESCR
FA_TYPE_DESCR	A60	CI_FA_TYPE_L.DESCR
DISP_GRP_DESCR	A60	CI_DISP_GRP_L.DESCR
LL_ACCT_ID	A10	CI_LANDLORD.ACCT_ID
LL_DESCR	A60	CI_LANDLORD_L.DESCR
MR_WARN_DESCR	A60	CI_MR_WARN_L.DESCR
MR_INSTR_DESCR	A60	CI_MR_INSTR_L.DESCR

Account / Person Information

Field Name	Format	Source/Value/Description
PREM_ID	A10	CI_FO
ACCT_ID	A10	CI_SA
MAIN_CUST_SW	A1	CI_ACCT_PER
PER_ID	A10	CI_ACCT_PER
ACCT_REL_TYPE_CD	A8	CI_ACCT_PER
ENTITY_NAME	A64	CI_PER_NAME
LS_SL_FLG	A2	CI_PER
LS_SL_DESCR	A254	CI_PER

Person Phone Information

Field Name	Format	Source/Value/Description
PER_ID	A10	CI_ACCT_PER - Primary person
SEQ_NUM	N3	CI_PER_PHONE
PHONE_TYPE_CD	A12	CI_PER_PHONE
COUNTRY_CODE	A3	CI_PER_PHONE
PHONE	A24	CI_PER_PHONE
EXTENSION	A6	CI_PER_PHONE
DESCR	A60	CI_PHONE_TYPE_L

Premise Geo Information

Field Name	Format	Source/Value/Description
PREM_ID	A10	CI_FO
GEO_TYPE_CD	A8	CI_PREM_GEO
GEO_VAL	A50	CI_PREM_GEO
GEO_TYPE_DESCR	A60	CI_GEO_TYPE_L.DESCR

SP Type Information

Field Name	Format	Source/Value/Description
SP_ID	A10	CI_FA
SP_DESCR	A254	CI_SP.DESCR254
SP_TYPE_CD	A8	CI_SP
SP_TYPE_DESCR	A60	CI_SP_TYPE_L.DESCR
SP_SUBTYPE_FLG	A2	CI_SP_TYPE
SP_SUBTYPE_DESCR	A60	CI_LOOKUP/SP Subtype Flag
SP_STATUS_FLG	A2	CI_SP
INSTALL_DT	Date	CI_SP
SP_SRC_STATUS_FLG	A2	CI_SP
DISCON_LOC_CD	A4	CI_SP
DISCON_LOC_DESCR	A60	CI_DISCON_LOC_L
MR_CYC_CD	A4	CI_SP
MR_CYC_DESCR	A60	CI_MR_CYC_L
MTR_LOC_CD	A4	CI_SP
MTR_LOC_DESCR	A60	CI_MTR_LOC_L
MTR_LOC_DETAILS	A250	CI_SP
FAC_LVL_1_CD	A8	CI_SP
FAC_LVL_1_DESCR	A60	CI_FAC_LVL_1_L.DESCR
FAC_LVL_2_CD	A8	CI_SP
FAC_LVL_2_DESCR	A60	CI_FAC_LVL_3_L.DESCR
FAC_LVL_3_CD	A8	CI_SP
FAC_LVL_3_DESCR	A60	CI_FAC_LVL_3_L.DESCR

SP Installed Meter Information

Field Name	Format	Source/Value/Description
SP_ID	A10	CI_FA
SP_MTR_HIST_ID	A10	CI_SP_MTR_HIST
MTR_CONFIG_ID	A10	CI_SP_MTR_HIST
EFF_DTTM	Date	CI_MTR_CONFIG
MTR_ID	A10	CI_MTR_CONFIG
INSTALL_DTTM	Date	CI_MR.READ_DTTM
READ_DTTM	Date	CI_MR
BADGE_NBR	A30	CI_MTR
MTR_TYPE_CD	A8	CI_MTR
MTR_STATUS_FLG	A2	CI_MTR
MFG_CD	A8	CI_MTR

MODEL_CD	A8	CI_MTR
SERIAL_NBR	A16	CI_MTR
RECEIVE_DT	Date	CI_MTR
MTR_DESCR	A254	CI_MTR.DESCR254
REG_ID	A10	CI_REG
READ_SEQ	N2	CI_REG
UOM_CD	A30	CI_REG
TOU_CD	A30	CI_REG
REG_CONST	N12.6	CI_REG
CONSUM_SUB_FLG	A2	CI_REG
HOW_TO_USE_FLG	A2	CI_REG
NBR_OF_DGTS_LFT	N2	CI_REG
NBR_OF_DGTS_RGT	N2	CI_REG
FULL_SCALE	N18.7	CI_REG
READ_OUT_TYPE_CD	A8	CI_REG
PROTOCOL_CD	A8	CI_REG
TOLERANCE	N14.5	CI_REG
REG_READ_ID	A12	CI_REG_READ
MR_ID	A12	CI_REG_READ
READ_TYPE_FLG	A2	CI_REG_READ
REG_READING	N15.6	CI_REG_READ
MTR_TYPE_DESCR	A60	CI_MTR_TYPE_L.DESCR
MFG_DESCR	A60	CI_MFG_L.DESCR
MODEL_DESCR	A60	CI_MODEL_L.DESCR

SP Installed Item Information

Field Name	Format	Source/Value/Description
SP_ID	A10	CI_FA
SP_ITEM_HIST_ID	A10	CI_SP_ITEM_HIST
ITEM_ID	A10	CI_SP_ITEM_HIST
INSTALL_DTTM	Date	CI_SP_ITEM_EVT.EVENT_DTTM
BADGE_NBR	A30	CI_ITEM
ITEM_TYPE_CD	A8	CI_ITEM
ITEM_STATUS_FLG	A2	CI_ITEM
MFG_CD	A8	CI_ITEM
MODEL_CD	A8	CI_ITEM
SERIAL_NBR	A16	CI_ITEM

RECEIVE_DT	Date	CI_ITEM
ITEM_DESCR	A254	CI_ITEM.DESCR254
ITEM_TYPE_DESCR	A60	CI_ITEM_TYPE_L.DESCR
MFG_DESCR	A60	CI_MFG_L.DESCR
MODEL_DESCR	A60	CI_MODEL_L.DESCR

SP Geo Information

Field Name	Format	Source/Value/Description
SP_ID	A10	CI_FA
GEO_TYPE_CD	A8	CI_SP_GEO
GEO_VAL	A50	CI_SP_GEO
GEO_TYPE_DESCR	A60	CI_GEO_TYPE_L.DESCR

SP Characteristics Information

Field Name	Format	Source/Value/Description
SP_ID	A10	CI_FA
CHAR_TYPE_CD	A8	CI_SP_CHAR
EFFDT	Date	CI_SP_CHAR
CHAR_VAL	A16	CI_SP_CHAR
ADHOC_CHAR_VAL	A254	CI_SP_CHAR
CHAR_VAL_FK1	A50	CI_SP_CHAR
CHAR_VAL_FK2	A50	CI_SP_CHAR
CHAR_VAL_FK3	A50	CI_SP_CHAR
CHAR_VAL_FK4	A50	CI_SP_CHAR
CHAR_VAL_FK5	A50	CI_SP_CHAR
CHAR_TYPE_FLG	A4	CI_CHAR_TYPE
FK_REF_CD	A8	CI_CHAR_TYPE
CHAR_TYPE_DESCR	A60	CI_CHAR_TYPE_L.DESCR

SP Multi Item Information

Field Name	Format	Source/Value/Description
SP_ID	A10	CI_FA

EFFDT	Date	CI_MULT_ITEM
ITEM_TYPE_CD	A8	CI_MULT_ITEM
ITEM_CNT	N11,2	CI_MULT_ITEM
ITEM_TYPE_DESCR	A60	CI_ITEM_TYPE_L.DESCR

FA Severance Process Information

Field Name	Format	Source/Value/Description
FA_ID	A10	CI_FA
SEV_PROC_ID	A10	CI_SEV_PROC
SA_ID	A10	CI_SEV_PROC
SEV_ARS_DT	Date	CI_SEV_PROC
ACCT_ID	A10	CI_SA

GL Accounting Summary - CI_GLACSM

Parameters

Parameter	Parameter Code	Description
Accounting Period	P_ACCT_PERIOD	Defines the accounting period used for the report. A valid fiscal year and accounting period for a valid accounting calendar must be provided.
Account Type Characteristic	P_CHAR_TYPE	Defines a Characteristic Type for a characteristic linked to the Distribution Code to define an Account Type. The account types for the GL accounts are used for grouping the output to the report.

Report Description

This is a financial audit report used to check the financial details in Oracle Utilities Customer Care and Billing for an accounting period against the GL system. The report summarizes all financial transaction (FT) information for a given accounting period according to the different operating and GL divisions and according to various levels of the account GL information.

NOTE:
Performance Consideration. If your implementation chooses to use this report, you may consider adding an index to the CI_FT table on ACCOUNTING_DT to aid in performance. When making this decision, carefully weigh the benefit of improving report performance against the possible degradation to the performance of day-to-day processing as a result of defining a new index. In addition, many companies opt

to create a reporting database that is a shadow of production to ensure that indexes defined to benefit reports may be created without any affect on the production environment.

Letter Print - BI Publisher Sample Welcome Letters - CI_LTRGN_ENG REPORT

NOTE:
Generating letters in batch is not supported yet.

Parameters

Parameter	Parameter Code	Description
Batch Switch	P_BATCH_SW	For reports accessed online, set the switch to 'No'. Otherwise, it should be set to 'Yes'.
Customer Contact	P_CC_ID	References the customer contact associated with the main customer linked to the account.

Report Description

This sample report template for BI Publisher produces letters that are not associated with any other object (i.e., the template does not have to extract information from another object to merge into a letter). You can use this template as a welcome letter for a new customer.

By default, the English versions of the report templates are provided with the base product. If multilingual report templates are required, your implementation should provide reports for each language. When a letter is generated, the system uses the report template based on the customer's language.

The address and name for the company are extracted from the installation options. The text for the letter is defined in the report layout and not provided by Oracle Utilities Customer Care and Billing. Reports are printed according to the customer's language definition and not based on the user's language definition.

This sample report uses the following text:

Welcome to %1. You have been filed with ID Number %2.

We hope to provide you with our best possible service. If you experience any problems or have any questions, please contact one of our customer service representatives at (800)1234567.

%1 is the company name stored as a message on the installation options.

%2 is Person Id stored on the Customer Contact.

Meter Reads Performance - CI_MTREAD

Parameters

Parameter	Parameter Code	Description

Accounting Period	P_ACCT_PERIOD	Defines the accounting period used for the report. A valid fiscal year and accounting period for a valid accounting calendar must be provided.

Report Description

This report selects bill segments for the input accounting period and displays the total number of read meters and unread meters for these bill segments grouped by route type.

Meters that are considered read are meters whose register reads have a status of Regular or Verified . Meters that are considered unread are meters whose register reads have a status of System Estimate, Office Estimate, System Prorate or Billing Force.

> **NOTE:**
> **Performance Consideration.** If your implementation chooses to use this report, you may consider adding an index to the CI_FT table on ACCOUNTING_DT to aid in performance. When making this decision, carefully weigh the benefit of improving report performance against the possible degradation to the performance of day-to-day processing as a result of defining a new index. In addition, many companies opt to create a reporting database that is a shadow of production to ensure that indexes defined to benefit reports may be created without any affect on the production environment.

Open Cases By Type - CI_CSEOPN

Parameters

Parameter	Parameter Code	Description
Start Date	P_FROM_DT	See the report's description for how this field is used. If start date is not specified, it is defaulted to 7 days prior to the end date.
End Date	P_TO_DT	See the report's description for how this field is used. If end date is not specified, it is defaulted to the current processing date.
Case Type	P_CASE_TYPE_CD	If specified, only cases of this type are included in the report.
Responsible User	P_CASE_OWNER	If specified, only cases with this responsible user are included in the report.
First Bucket High Limit	P_B1_LIMIT	Cases that are open for less than or equal to this number of days will be grouped together for statistical reporting.
Second Bucket High Limit	P_B2_LIMIT	Cases that are open less than or equal to this number of days but more than the first bucket high limit will be grouped together for statistics reporting.
Third Bucket High Limit	P_B3_LIMIT	Cases that are open less than or equal to this number of days but more than the second

		bucket high limit will be grouped together for statistics reporting. Cases that took more than this number of days are included in another group.

Report Description

This is a report on open cases that were created between a given date range.

The report can be limited to a specific type and/or responsible user.

For each case type, the report shows the following:

- Number of open cases by age bucket (the last 3 parameters control the size (in days) of each bucket)
- Percentage of open cases by age bucket
- Details of the open cases

Payments Balance - CI_PMTBAL

Parameters

Parameter	Parameter Code	Description
Start Date	P_FROM_DT	Report gets all the payments that have been received during a given date range (from and to date parameters). If start date is not defined by the user, it is set to 7 days prior to the current date.
End Date	P_TO_DT	End date of the date range. If not defined by user it is set to the current date

Report Description

This report provides an overall view of all payments created within the input date range. It is typically used for financial control and audit purposes. The report provides summary information about valid payments received and about canceled payment. Data is summarized by the tender source and the type of payment.

Receivables Aging - CI_RCVAGA

Parameters

Parameter	Parameter Code	Description
Cutoff Date	P_CUTOFF_DATE	The date from which the arrears buckets are calculated. If no value is entered, the default is the current date minus 7 days.
First Bucket High Limit	P_B1_LIMIT	High limit of first bucket.
Second Bucket High Limit	P_B2_LIMIT	High limit of second bucket.
Third Bucket High Limit	P_B3_LIMIT	High limit of third bucket.

Report Description

This report lists all accounts and their arrears information as of the input cutoff date using a balance forward accounting method.

Outstanding debt is placed into the buckets provided as input using the age of the debt as of the cutoff date. Credits are applied to the oldest debt first. For each account a separate bucket is used to display new charges. In addition, the total accounts receivable balance is displayed for each account.

NOTE:
Performance Consideration. If your implementation chooses to use this report, you may consider adding an index to the CI_FT table on ARS_DT to aid in performance. When making this decision, carefully weigh the benefit of improving report performance against the possible degradation to the performance of day-to-day processing as a result of defining a new index. In addition, many companies opt to create a reporting database that is a shadow of production to ensure that indexes defined to benefit reports may be created without any affect on the production environment.

Tax Payables Analysis - CI_TXPYBL

Parameters

Parameter	Parameter Code	Description
Start Date	P_FROM_DT	Show summary of the tax amounts starting from this date. If not specified, the system will default this value to the current date minus 7 days.
End Date	P_TO_DT	Show summary of the tax amounts up to this date. If not specified, the system will default this value to the current date.
Account Type Characteristic	P_CHAR_TYPE	Defines a Characteristic Type for a characteristic linked to the Distribution Code to define an Account Type.
Account Type	P_TAX_ACCTY_CHAR	Account Type Char Value for tax related GL Accounts. The char type defined for this parameter should match the Char Type code defined as parameter #3. The parameter value indicated for this parameter should be one that represents tax liability accounts.

Report Description

This report displays a summary of the tax amounts that were levied by the company to customers within the input date range. It also includes the tax exemption information for that period.

This report select all records in the financial transaction GL collection that satisfy the following criteria:

- The financial transaction is frozen.
- The Accounting Date on the financial transaction within the input date range
- The distribution code associated with the GL entry has a characteristic type and value that matches the input Account Type Characteristic and Account Type (parameters 3 & 4)

The report also provides tax exemption information for bill segments whose financial transactions satisfy the above criteria. The tax exemption information is retrieved by looking at the bill calculation lines associated with the FT's bill segment.

➤ **NOTE:**

Performance Consideration. If your implementation chooses to use this report, you may consider adding an index to the CI_FT table on ACCOUNTING_DT to aid in performance. When making this decision, carefully weigh the benefit of improving report performance against the possible degradation to the performance of day-to-day processing as a result of defining a new index. In addition, many companies opt to create a reporting database that is a shadow of production to ensure that indexes defined to benefit reports may be created without any affect on the production environment.

To Do Entries - CI_TDENTR

Parameters

Parameter	Parameter Code	Description
ToDo Entry Status	P_ENTRY_STATUS_FLG	Defines if the To Do entries on the report should be limited to those with a given status value. If this parameter is left blank, the report will show all open and being worked To Do entries.
ToDo Type	P_TD_TYPE_CD	Defines if the To Do entries on the report should be limited to those of a given ToDo type. If this parameter is left blank, the report will show all ToDo type that have at least one open or being worked entry.

Report Description

The report shows open and being worked To Do entries.

You can limit the report to entries in a given status by specifying the desired **To Do Status** (open or being worked). If you don't specify a status, all open and being worked To Do entries will appear on this report.

You can also limit the report to entries of a given To Do Type by specifying the desired To Do Type. If you don't specify a To Do Type, all To Do Types with at least one entry in the open / being worked state will appear on this report.

➤ **NOTE:**

Graphs. The information on this report is shown in both textual and graphical formats.

Umbrella Agreement Summary - CI_UASUMM

Parameters

At least one parameter is required. If the umbrella agreement is entered, no other parameter value is allowed. The report definition uses the validation algorithm RPTV-UASUMM to check these conditions.

Parameter	Parameter Code	Description

Umbrella Agreement	P_UA_ID	Specify the Umbrella Agreement to restrict the report to this umbrella agreement (regardless of its status). When this parameter is specified, all other parameters are not allowed.
Account Management Group	P_AMG	Specify a value for this parameter to restrict the report to umbrella agreements related to this Account Management Group.
Umbrella Agreement Characteristic Type	P_CHAR_TYPE	Specify a value for this parameter to restrict the report to completed umbrella agreements related to this Characteristic type and value (parameter 4).
Umbrella Agreement Characteristic Value	P_CHAR_VALUE	Specify a value for this parameter to restrict the report to completed umbrella agreements related to this Characteristic type (parameter 3) and value.
Number of Days Before Expiration	P_EXPIRE_IN_X_DAYS	Specify a value for this parameter to restrict the report to completed umbrella agreements whose End Date is on or before the current date less the Number of Days before Expiration.

Report Description

This report provides summary information about one or more umbrella agreements.

If the user does not supply an umbrella agreement ID, but enters one or more of the other parameters, the report only selects umbrella agreements that match the input criteria AND the match the following criteria:

- The status of the umbrella agreement is complete
- The start and end dates of the umbrella agreement include the current date

The report does not include canceled terms of service records or canceled service agreements linked to the above umbrella agreements.

For each umbrella agreement that is included in the report, the output includes information about the umbrella agreement, its collection of terms of service records and for each TOS record, its collection of service agreements.

In addition, the report includes the following graphs:

- Umbrella agreement summary graph showing billing history by accounting period for the effective dates of the umbrella agreement for all service agreements linked to the umbrella agreement's terms of service records.
- Terms of service summary graph showing billing history by accounting period for the effective dates of the umbrella agreement for all its service agreements

Unbilled Revenues - CI_UBLREV

Parameters

Parameter	Parameter Code	Description
Accounting Period	P_ACCT_PERIOD	Defines the accounting period used for the report. A valid fiscal year and accounting

		period for a valid accounting calendar must be provided.
Account Type Characteristic	P_CHAR_TYPE	Defines a Characteristic Type for a characteristic linked to the Distribution Code to define an Account Type.
Account Type	P_REV_ACCTY_CHAR	Account Type Char Value for Revenue related GL Accounts. The char type defined for this parameter should match the Char Type code defined as parameter #2. The parameter value indicated for this parameter should be one that represents revenue accounts.

Report Description

This report provides a simplified calculation of estimated unbilled revenue for a given month.

It processes frozen financial transactions associated with bill segments and canceled bill segments whose bill segment end date is within the **Accounting Period** (parameter 1). It determines the billed revenue for the FT and then uses this information to calculate the unbilled revenue.

To determine "revenue", the report summarizes amounts posted to any distribution code with a characteristic entry that matches the input **Account Type Characteristic** type and **Account Type** characteristic value (parameters 2 and 3).

The estimate for the unbilled portion is calculated as (Bill Amount / Number of Days in the Bill Period) * Number of Unbilled Days for the Accounting Period.

For example, consider a bill segment on a monthly-billed cycle for the period of 3/10/03 - 4/9/03, and for $150 (revenue part of the bill). Assume we are in the April/03 accounting period, which covers 4/1/03 through 4/30/03. This means 21 days are unbilled for April. The unbilled revenue is calculated as 150/31*21 = $101.61.

The following are some points to note about this report.

- If the report runs for an historical date, it still estimates the unbilled revenue portion based on the above formula even if actual data exists. In other words, it will not try to find actual bills for subsequent months. As a result, this report always shows what would have been the estimated unbilled revenue for a particular month if that month is the most recent one.
- Services can be billed monthly bi-monthly, quarterly or in any desired frequency. This report only performs the revenue recognition estimation using actual bills that were created in the report's accounting period. As a result, it does not estimate unbilled revenue for accounts that were not billed in any portion of the input accounting period.
- Revenue recognition practices are unique and may vary from customer to customer. In a given month we can produce current bills (e.g. for electric service) or future bills (e.g. cable services). In addition in some cases we can have bills that started before the accounting period and ending after it (for example in a cancel-rebill situation). This report will only estimate unbilled revenues for bills with the accounting period equals to the report's accounting period.

> **NOTE:**
> **Performance Consideration.** If your implementation chooses to use this report, you may consider adding an index to the CI_BSEG table on START_DT, END_DT to aid in performance. When making this decision, carefully weigh the benefit of improving report performance against the possible degradation to the performance of day-to-day processing as a result of defining a new index. In addition, many companies opt to create a reporting database that is a shadow of production to ensure that indexes defined to benefit reports may be created without any affect on the production environment.

Vacant Premises with Consumption - CI_VACANT

Report Description

This report shows all premises that are considered vacant, and provides information about the level of consumption, the period of vacancy, and the service point and register information.

This report selects service points that are linked to a service agreement that is either canceled or closed. The report excludes any service points that have never been linked to a service agreement. If consumption has been registered at such a service point since the end date of the service agreement, the service point's details are displayed.

This report may be used by the utility to investigate sites where problems such as service theft or leakage may be occurring.

> **NOTE:**
> **Performance Consideration.** If your implementation chooses to use this report, you may consider adding an index to the CI_SA_SP on STOP_DT to aid in performance. When making this decision, carefully weigh the benefit of improving report performance against the possible degradation to the performance of day-to-day processing as a result of defining a new index. In addition, many companies opt to create a reporting database that is a shadow of production to ensure that indexes defined to benefit reports may be created without any effect on the production environment.

Defining Overdue Processing Options

The system periodically monitors how much your customers owe to ensure they haven't violated your collection rules. When a violation is detected, the system initiates the appropriate activities (e.g., letters, disconnect notices, collection agency referrals, and eventually write off). The topics in this section describe how to configure the system to manage your overdue processing requirements.

➤ **NOTE:**
The overdue processing module has been designed to collect on virtually anything from an unpaid bill to an unmatched financial transaction. You tell the system what you collect on by configuring the various overdue processing control tables. In this release, the base-package is delivered with algorithms that support collecting on overdue bills. If your organization practices balance-forward accounting (i.e., collection is based on overdue service agreement balances), you will not use this functionality. Rather, you will use the functionality described under Defining Credit and Collections Options.

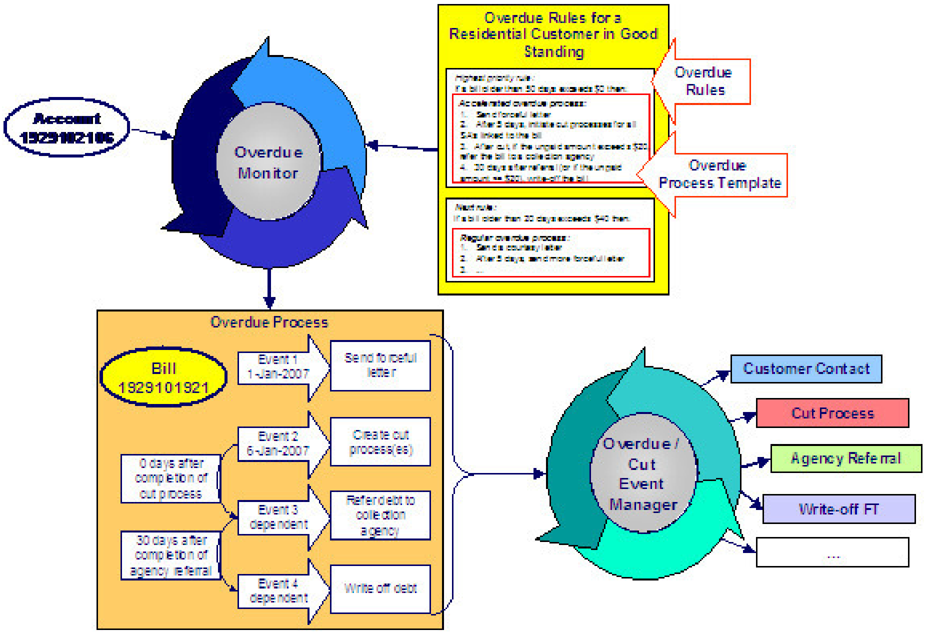
WARNING:
Setting up this module is as challenging as your organization's collection rules. If you have simple rules, the set up process will be straightforward. If your rules are complicated (e.g., they differ based on the type of customer, the age of debt, the type of service, etc.), your setup process will be more challenging.

Case Study - Collecting On Overdue Bills

The following topics introduce a case study that describes how overdue processing can be used to collect on overdue bills. This is just an example as virtually every aspect of overdue processing is configurable. Use this case study to familiarize yourself with the overdue processing core concepts.

Monitoring Overdue Bills

The following diagram illustrates the objects and processes involved with collecting overdue bills.


Account
1929102106
Overdue Monitor
Overdue Rules for a Residential Customer in Good Standing
Highest priority rule:
If a bill older than 30 days exceeds $0 then:
Accelerated overdue process:
1. Send forceful letter
2. After 5 days, initiate cut processes for all SAs linked to the bill
3. After cut, if the unpaid amount exceeds $20, refer the bill to a collection agency
4. 30 days after referral (or if the unpaid amount <= $20), write-off the bill
Next rule:
If a bill older than 20 days exceeds $40 then:
Regular overdue process:
1. Send a courtesy letter
2. After 5 days, send more forceful letter
3. ...
Overdue Rules
Overdue Process Template
Overdue Process
Bill
1929101921
Event 1
1-Jan-2007
Send forceful letter
Event 2
6-Jan-2007
Create cut process(es)
0 days after completion of cut process
Event 3
dependent
Refer debt to collection agency
30 days after completion of agency referral
Event 4
dependent
Write off debt
Overdue / Cut Event Manager
Customer Contact
Cut Process
Agency Referral
Write-off FT
...

There are many important concepts illustrated above:

The Overdue Monitor checks if your accounts have bills that violate your overdue rules	The Overdue Monitor is a background process that periodically reviews your account's financial obligations. Note well: every account belongs to a collection class. There are two types of collection classes: those whose accounts are monitored by the Account Debt Monitor, and those that are monitored by the Overdue Monitor. This chapter describes the Overdue Monitor.
Overdue rules define when and how unpaid bills are collected	An account's collection class overdue rules have algorithms that monitor an account's financial obligations. These algorithms are invoked by the Overdue Monitor when it's time to review an account's obligations. These algorithms can contain any type of criteria. However, most are defined using a combination of a threshold age and monetary amount. For example, a classic algorithm would check if a bill has unpaid financial transactions more than 20 days old that exceed $50. In the case of bill-oriented collection, the monitoring algorithms look at each of the account's bills to determine if they are paid. Note, a bill is considered paid if its financial transactions (FTs) are linked to a balanced match event. If a monitoring algorithm finds an unpaid bill, it can check if it old enough (and large enough) to be considered a violation. When you set up a monitoring algorithm, you define the type of overdue process that should be created when an overdue bill is detected. You do this by defining the appropriate "overdue process template".
An overdue process template defines how to handle an overdue bill	An overdue process template contains one or more overdue event types. These define the number and type of events that are created to prod the customer to pay. For example, you might set up an overdue process template with event types to send a series of letters followed up by a call.

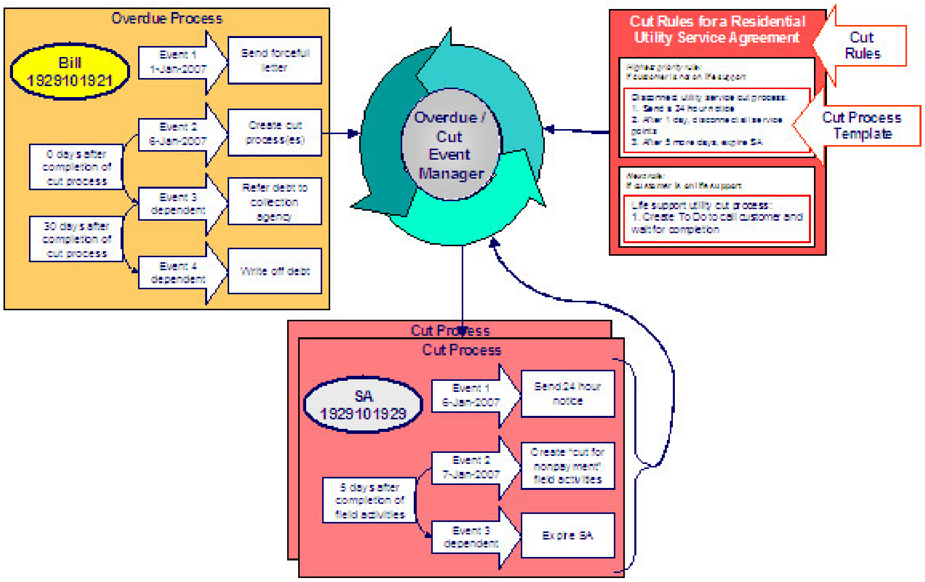
	The overdue process template has contains the rules defining when events are activated. The specific action that's performed by an overdue event is controlled by the Activation algorithm defined on its event type. Refer to Overdue Event Type - Main for a list of the various Activation algorithms delivered with the base package.
Multiple objects can be associated with a single process	The above diagram shows a single bill linked to an overdue process. It should be noted that an overdue process is capable of referencing multiple bills (or other objects). Note well: while a single overdue process can reference many overdue objects, all such objects must be of the same type. For example, you cannot commingle bills and service agreements under a single overdue process. The type of object managed by an overdue process is defined on its overdue process template.
If a customer pays the bill, the overdue process is cancelled	If an overdue bill is paid, the overdue process is canceled real-time. You control if and how an overdue process is cancelled by setting up the appropriate rules on the overdue process template.
The Overdue / Cut Event Manager activates and triggers overdue events	The Overdue / Cut Event Manager is a background process that activates overdue events on the appropriate date. On this date, the event's Activation algorithm(s) are called. This Overdue / Cut Event Manager also has the responsibility of recursively activating later events that are dependent on the completion of earlier events.
Events can be activated real-time	Overdue Process - Main has a button that allows users to activate (and recursively trigger) overdue events online / real-time. This means you don't have to wait for a batch job to activate events.
Overdue events can wait for related activities to complete	As described above, an overdue event's Activation algorithm can create virtually any object. What wasn't explained is that the event can be set up to wait for the ancillary object to finish before it completes. For example, an event can create a To Do entry and wait for it to complete before the next event is triggered. You can introduce plug-ins to create and wait on virtually any object. While an overdue event is in the Wait state, the Overdue / Cut Event Manager monitors the state of the related object(s). When the related object completes, the event is transitioned to the Complete sate (thus triggering dependent overdue events). Please see Some Events Wait For Something Before Completing for more information.

Cutting Service Agreements

An overdue process may contain an overdue event that creates a cut process(es). A cut process is used to cut (i.e., stop) a service agreement. The following diagram illustrates the objects and processes involved with cutting a service agreement.

➤ **NOTE:**

Cut processes aren't required. It's quite conceivable to design an overdue process that doesn't cut service. For example, the overdue process may just contain an event that creates a case and the case manages the collection activities.


Overdue Process
Bill
1929101921
Event 1
1-Jan-2007
Send forceful letter
Event 2
6-Jan-2007
Create cut process(es)
0 days after completion of cut process
Event 3
dependent
Refer debt to collection agency
30 days after completion of cut process
Event 4
dependent
Write off debt
Overdue / Cut Event Manager
Cut Rules for a Residential Utility Service Agreement
Highest priority rule:
If customer is not on life support
Disconnect utility service cut process:
1. Send a 24 hour notice
2. After 1 day, disconnect all service points
3. After 5 more days, expire SA
Next rule:
If customer is on life support
Life support utility cut process:
1. Create To Do to call customer and wait for completion
Cut Rules
Cut Process Template
Cut Process
Cut Process
SA
1929101929
Event 1
6-Jan-2007
Send 24 hour notice
Event 2
7-Jan-2007
Create "cut for nonpayment" field activities
5 days after completion of field activities
Event 3
dependent
Expire SA

An overdue process may contain an overdue event that cuts (i.e., stops) service	If an overdue process's bill remains unpaid, one of the latter overdue events typically creates one or more cut processes. A cut process contains the activities to stop a service agreement (in the hopes that lack of service will inspire the customer to pay). It should be noted that a separate cut process is created for each service agreement.
Cut rules define how to sever service agreements	The system allows you to define rules to control the type of cut process created. For example, you may have a different cut process if the customer has life support equipment, or if it's winter, or ... The cut rules are defined on the service agreement's SA type. This allows different rules for different types of service agreements.
Overdue events can "wait" for related activities to complete	When an overdue event creates cut process(es), it typically enters the Wait state. It enters this state as it is waiting for the cut process(es) to complete before it can itself complete. While in the Wait state, the Overdue / Cut Event Manager monitors the state of the related cut process(es). When the cut process(es) complete, the system transitions the originating overdue event to the Complete state (thus triggering dependent events).
A cut process template defines how to cut a service agreement	Cut processes and events are created using a cut process template. A cut process template defines the actions involved in cutting a given type of service agreement. A cut process template usually contains several cut events. These events are a series of letters and / or disconnection field activities that eventually result in the expiration of a service agreement if payment is not received. An Activation algorithm controls the specific action associated with a cut event and therefore an event can do practically anything. Refer to

	Cut Event Type - Main for a list of the various Activation algorithms delivered with the base package.
Cut events are also managed by the Overdue / Cut Event Manager	The same background process that manages the activation of overdue events manages cut events. This means you only have to submit one batch job to activate and trigger both overdue and cut events.
Events can be activated real-time	Cut Process - Main has a button that allows users to activate (and recursively trigger) cut events online / real-time. This means you don't have to wait for a batch job to activate events.
Cut events can "wait" just like overdue events	Cut events support the ability to wait for something to complete just like overdue events. For example, if a cut event creates a field activity, it enters the Wait state. While in the Wait state, the Overdue / Cut Event Manager monitors the state of the related field activities. When the field activities complete, the system transitions the originating cut event to the Complete state. And, just like for overdue events, the notion of a cut event creating something and then waiting for it to complete is not limited to field activities. For example, you could have a cut event create a To Do entry and wait for it to complete before the next event is triggered. Please see Some Events Wait For Something Before Completing for more information.
After a service agreement is stopped, it will be final billed	Many cut processes are configured so their last cut event expires the service agreement. What happens at expiration depends on the type of service agreement. However, eventually the service agreement is transitioned into the Stopped state. When the last active service agreement linked to an account is stopped, the system changes the account's bill cycle to bill that evening. If only one of many SAs is stopped, the SA will only be final billed as per the account's original bill cycle schedule.
If a customer doesn't pay their final bill, the final bill will be processed using a different type of overdue process	Final bills only differ from ongoing bills in that the service agreements associated with the bill's FTs are not active. This means that there is nothing to cut. This means that the type of overdue process used to handle a "final" bill should differ from that used to handle "ongoing" bills. As described above, overdue rules define how unpaid bills are handled (both ongoing and final). We recommend configuring your overdue rules to use different overdue process templates for final versus ongoing bills.
The last overdue event typically causes the debt to be written off	Ultimately, if the overdue and cut events fail to inspire payment, the debt will be written off. The method used to write-off a bill is controlled by an overdue event Activation algorithm.

How Does The Overdue Monitor Work?

This section describes how the Overdue Monitor background process (batch control: C1-ADMOV) uses your overdue rules to collect overdue debt. If your implementation requires processing of data by CIS Division the background process (C1-OAM) is provided that allows a CIS Division to be specified as an input parameter.

NOTE:
Recommendation. We recommend that you familiarize yourself with the concepts described in the case studies before reading this section.

Collecting overdue bills. The examples in the following section frequently refer to how the Overdue Monitor is configured for an organization that collects on unpaid bills. It should be noted that these are just examples as the Overdue Monitor can be used to collect on virtually anything (if you create the appropriate plug-in algorithms).

Different Overdue Rules For Different Customers

The Overdue Monitor uses rules to control how it monitors an account's debt. The system allows you to define different rules for different combinations of collection class, division and currency code. For example,

- You probably have different collection rules for different jurisdictions (i.e., CIS Divisions). For example, if you have customers in different states / provinces, you may have different rules applied to each jurisdiction's debt. The **CIS division on each customer's account** defines the jurisdictional rules applied to the account's debt.
- You probably have different collection rules for different customer segments. For example, bills for large customers might be overdue if they are more than 10 days late, whereas small customers might have 24 days. You differentiate your customers in respect of the overdue via the **collection class on the customers' accounts**. An account's initial collection class is defaulted from its customer class. You may override an account's collection class at will.
- You will have different criteria for every currency in which you work because the monitoring process always compares a customer's debt against some value and this value must be denominated in the customer's currency. A customer's currency is defined using a **currency code on the account**.

The above means that every combination of CIS division, collection class, and currency code can have different rules. The following matrix is used to illustrate a sample organization's rules (note, we assume a single currency and therefore avoid the third dimension):

Account's Division	**Account's Collection Class:**	**Account's Collection Class:**
	Commercial Customer	Residential Customer
North	Highest Priority: If a bill exists with unpaid FT's > $0 that is older than 45 days, create the commercial 45 days late overdue process. Next Priority: If a bill exists with unpaid FT's > $100 that is older than 30 days, create the commercial 30 days late overdue process.	Highest Priority: If a bill exists with unpaid FT's > $0 that is older than 50 days, create the accelerated overdue process for residential customers. Lower Priority: If a bill exists with unpaid FT's > $25 that is older than 25 days, create the courtesy reminder overdue process for residential customers.
South	...	...

Notice that there can be multiple criteria for each cell in the matrix. What differentiates one criteria from another is its priority. The higher priority criteria will be compared first. If the debt violates the criteria, the overdue process is initiated and no further comparisons are performed.

Overdue Rules Are Embodied In Algorithms

Your organization's overdue rules are defined in algorithms plugged in on Collection Class Overdue Rules (in the Overdue Monitor Rule system event). When the Overdue Monitor analyzes an account, it uses the rules associated with the account's collection class, CIS division and currency code. To analyze an account, it simply invokes these algorithms in sequence order, i.e., the lower the sequence, the higher its priority.

An Overdue Monitor Rule algorithm has two responsibilities:

- it determines if an account violates its overdue rules,
- if so, it creates one or more overdue processes using an overdue process template

When Is An Account Monitored?

The Overdue Monitor determines if an account violates your overdue rules at least every X days, where X is defined on the Customer Class - Controls associated with the account's customer class and division (in the field Min Credit Review Freq (Days)).

In addition, the Overdue Monitor examines an account's debt as follows:

- X days after an account's bill due date (X is defined in the field Credit Review Grace Days on customer class control).
- If a payment is canceled with a cancellation reason that indicates non-sufficient funds.
- If a payment arrangement is broken (assuming you use the base package's break payment arrangement plug-in). Refer to Breaking A Bill Oriented Payment Arrangement for more information.
- If a pay plan is broken. Refer to The Pay Plan Monitor for more information.
- If a match event is added, changed or deleted.
- When a written off bill is reversed.

➤ **NOTE:**
Additional events. Your implementation can have other events trigger the analysis of an account by the Overdue Monitor. To do this, add logic to insert a row on the CI_ADM_RVW_SCH table when the event occurs. This row simply references the account ID to be reviewed and the desired review date.

Collection Class Defines If And How Accounts Are Monitored

As described above, every account references a collection class. The collection class defines if and how its accounts are monitored. There are three options:

- The accounts are monitored by the Account Debt Monitor
- The accounts are monitored by the Overdue Monitor (this is described in this chapter).
- The accounts are not monitored for overdue debt.

➤ **NOTE:**

> **Migration.** If you plan to migrate from the Account Debt Monitor (ADM) method of collection to the Overdue Monitor method, a special Overdue Monitor algorithm (C1-ACT-CSW) has been supplied that will skip accounts that are eligible for review by the Overdue Monitor if they have an active collection, severance or write-off process. This algorithm should be plugged in the applicable Collection Class Overdue Rules so that it will be invoked first. This allows accounts with active ADM-oriented collection activities to work themselves through the system. When an account no longer has any active ADM-oriented activities, monitoring responsibilities will be assumed by the Overdue Monitor.

The Big Picture Of Overdue Processes

As described above, the Overdue Monitor subjects your accounts to overdue rules. If a rule is violated, an overdue process is created. The topics in this section provide background information about overdue processes.

How Are Overdue Processes Created?

As described above, the system creates an overdue process when an account violates your overdue rules. In addition, a user can manually create an overdue process at their discretion.

The Components Of An Overdue Process

The following topics describe the major components of an overdue process.

Overdue Objects

When an overdue process is created, the system links the overdue object(s) to the process. For example, if an overdue bill is detected, the bill is linked to the overdue process.

When you set up an overdue process template, you define the type of object it collects on by defining the foreign key characteristic type used to reference the object. For example, when you set up an overdue process template to collect on bills, you define a foreign key characteristic type that references the bill object.

Overdue Events

An overdue process has one or more overdue events. These events are the actions designed to encourage the customer to pay. For example, you might set up overdue events that:

- Send letters (via the creation of a customer contact)
- Create To Do entries
- Impact the account's credit rating
- Create a case (e.g., to seize the customer's assets)
- ... (the list is only limited by your imagination as algorithms are used to perform the event's actions)

You define the number and type of events by configuring overdue process templates. When the system creates an overdue process, it copies the events defined on the specified template.

It's important to note that all overdue events are created when the overdue process is created. A separate background process, the Overdue / Cut Event Manager, is responsible for activating, monitoring, and triggering overdue events. Activation of an event causes the system to do whatever the event indicates (e.g., send a letter, send a To Do entry to a user, start a cut process, refer debt to a collection agency, write-off debt, etc.).

Overdue Log

Every overdue process has a log holding its history. Entries are added to the log when meaningful events occur, for example:

- When the process is created, a log entry is created to describe why the process was started.
- When an overdue event is activated, a log entry is created. These entries frequently contain a foreign key to the object that the event created so that users can easily drill down to the object from the log. For example, if an event creates a To Do entry, the To Do entry's foreign key is placed on the log and this allows a user to drill down on the log entry to see the To Do entry.
- When a process is canceled, a log entry is created to describe the circumstances of the cancellation (e.g., manual versus automated).
- Users can manually add log entries (you might want to think of these as "diary" entries) as desired.
- ...

Many of the base-package algorithms involved in overdue processing insert log entries so that a thorough audit trail is maintained. These algorithms have been designed to allow you to control the verbiage in each log entry by defining the desired message number using an algorithm parameter.

The log is viewable on the Overdue Process - Log page.

NOTE:
More than just an audit trail. The log entries are more than just an audit trail. The system makes use of the log entries to know what it did. For example, when an overdue event needs to monitor the state of the To Do entries that it created, it uses the log to determine the identity of these To Do entries.

Experimenting With Alternative Overdue Process Templates

The system allows you to determine the efficacy of proposed overdue process templates using a small subset of customers before implementing the templates on the entire customer base. We use the term "champion / challenger" to reference this functionality.

We'll use an example to explain. Let's assume your prevailing overdue process template for residential customers starts with a "gentle reminder" letter followed 10 days later by a letter threatening collection agency referral if payment is not received. You may want to experiment with the impact of a change to this template. For example, you may want to change the "gentle reminder" to something more assertive and follow this up 5 days later with an even sterner warning. You can use the "champion / challenger" functionality to perform this experiment.

The following points describe how to implement "champion / challenger" functionality:

- Set up a "challenger" overdue process template for each template that you want to experiment with.

- Insert a new **Champion/Challenger** option on the Overdue Processing Feature Configuration for every champion template. Each option's value defines:
  - the "champion" overdue process template code
  - the "challenger" overdue process template code
  - the percentage of the time the system should use the "challenger" template
- Keep in mind that you can only experiment with one challenger template per champion template. For example, let's assume you have two prevailing overdue process templates - one for residential customers and another for commercial customers. You can experiment with different challenger templates for the residential and commercial templates. However, you cannot experiment with two different challenger templates for the residential champion template (i.e., a champion template can have 0 or 1 challenger template).

After setting up the above, your implementation's Overdue Rule Plug-In may include logic to use the challenger template X % of the time rather than the champion template. The sample plug-in provided in the base product, called C1-CB-CR-RAT, includes this logic.

If you are using the Oracle Utilities Analytics product, you can configure analytic zones in innumerable ways to compare the efficacy of the champion versus the challenger. For example,

- You can set up a graph to show the average duration of each type of process.
- You can set up a graph to show the average dollars that were successfully collected.
- You can set up a dimensional scorecard to show how each template performed in different regions (or customer classes or ...).
- Etc (the list is limited by your imagination)

Overdue Process Information Is Overridable

"Overdue process info" is the concatenated string of information that summarizes an overdue process throughout the user interface. The base-package logic constructs this string by concatenating the following information:

- The description of its overdue process template
- Its status
- For active processes, the number of days since it was created. For inactive processes, the number of days since it was inactivated.
- For active processes, the unpaid amount of the objects being collected

If you'd prefer a different info string, you can develop a new algorithm and plug-it in on your overdue process templates. This design allows some / all overdue process templates to have an override info string.

Original and Unpaid Amounts

There are two amounts associated with each overdue object linked to an overdue process: its Original Amount and its Unpaid Amount. These amounts are used throughout overdue processing.

You control how these amounts are calculated by defining the appropriate algorithm on your overdue process templates. For example, you can plug in a base-package algorithm (C1-CUAOA) if you collect on overdue bills and the original amount is the amount of financial transactions linked to the bill, and the unpaid amount is the sum of financial transactions that are not linked to balanced match events.

Overdue Processes Are Highlighted Elsewhere

The topics in this section describe how other parts of the system highlight the existence of overdue and cut processes.

Alert Zone

If you plug-in the appropriate alert algorithm (C1-OD-PROC) on the Installation record, alert(s) will be shown for active overdue processes in the Alert Zone that appears in the Dashboard and on Control Central - Account Information.

Credit and Collections Zone

The Credit and Collections zone on Control Central - Account Information shows active overdue and cut processes.

Account Activity History Zone

The Account Activity History Zone on Control Central - Account Information shows pending and waiting events and inactive processes.

How Are Overdue Processes Cancelled?

A user may cancel an overdue process at their discretion, online / real-time using Overdue Process - Main.

The system will automatically cancel an overdue process when the object(s) associated with the overdue process are sufficiently paid. Exactly when the system checks if an overdue process should be cancelled is dependent on your organization's billing and accounting rules. For example, if you practice open-item accounting, you'd want to analyze an account's active overdue processes whenever a match event is added, changed or deleted (as match events are the only objects that impact if debt is considered paid in an open-item world). The base-package supports this specific example. If you need additional events to check if an overdue process should be canceled (e.g., the creation of a pay plan), a base-package change MAY be necessary. Please check with customer support if you have questions.

Two algorithms plugged-in on the overdue process template handle the cancellation:

- The Cancel Criteria algorithm is responsible for determining if an overdue process should be canceled. Algorithms of this type analyze the outstanding debt on the objects linked to the overdue process and indicate whether a process can be cancelled.
- The Cancel Logic algorithm is responsible for actually canceling the process. The logic involved in cancellation can be quite sophisticated as canceling an overdue process can result in the cancellation of its pending events and cut processes.

NOTE:
Why two algorithms? The reason two algorithms are involved in cancellation is because we want the cancellation logic to be encapsulated in one place so it can be called during both manual and automated cancellation.

Different logic for different templates. Because both the Cancel Criteria and Cancel Logic algorithms are plugged-in on the overdue process's template, you can have different cancellation criteria and logic for different templates.

Overdue Processes Are Created From Templates

As described above, you set up overdue process templates to define the types of events and when they are executed. When an overdue process is created, its events are created by copying the event types from an overdue process template. The remaining topics in this section provide background information to assist you in setting up your templates.

The Big Picture Of Overdue Events

This section describes the various types of overdue events and their lifecycle.

How Are Overdue Events Created?

Overdue events are created as follows:

- The Overdue Monitor invokes Overdue Monitor Rules to periodically check your accounts (refer to Overdue Rules Are Embodied In Algorithms for how this works). An Overdue Monitor Rule creates an overdue process when an account violates your overdue rules. The overdue process has one or more overdue event(s). The number and type of events is controlled by the overdue process template specified on the Overdue Monitor Rule.
- Users can create an overdue process manually on Overdue Process - Main. To do this, they specify an overdue process template. The number and type of overdue events is defaulted from the template.
- An overdue event may be manually added to an existing overdue process by a user on Overdue Process - Events.

➤ **NOTE:**
Bottom line. Most overdue events are created by the system when it creates an overdue process for delinquent obligations. If you need to create an ad hoc overdue event, you can either create an overdue process whose template contains the desired event OR link the desired event to an existing process.

Overdue Events Can Do Many Things

An overdue event can perform a wide number of activities as the logic is embodied in an algorithm. The following points describe how this works:

- Every overdue event references an overdue event type.
- The overdue event type, in turn, references an Event Activation algorithm.
- The Event Activation algorithm is invoked when the event is triggered.

Overdue Event Information Is Overridable

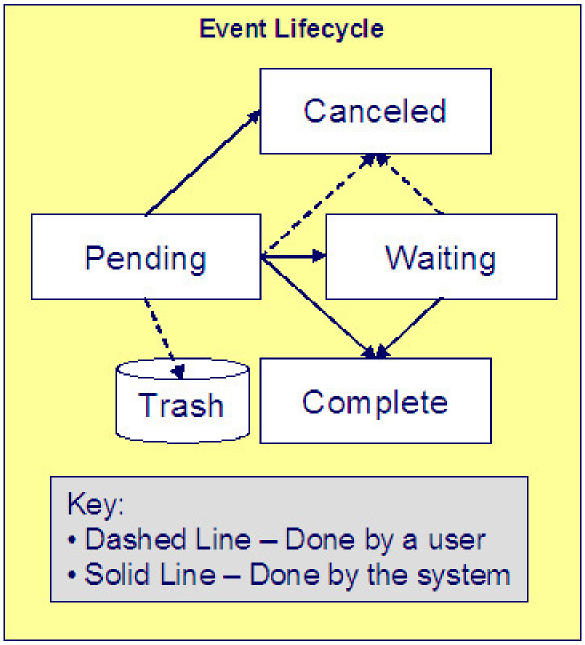
"Overdue event info" is the concatenated string of information that summarizes an overdue event throughout the system. The base-package logic constructs this string by concatenating the following information:

- The event type's description
- The event's status
- If it's pending:
  - If the event has a trigger date, the number of days until it's triggered plus the verbiage day(s) from today
  - Otherwise, the verbiage dependent on other events
- If it's waiting, the number of days, hours and minutes that it's been waiting
- If it's canceled, the cancel reason code's description
- If it's complete, the number of days, hours and minutes that it's been complete

If you'd prefer a different info string, you can develop a new algorithm and plug-it in on your event types. This design allows some / all event types to have an override info string.

Overdue Event Lifecycle

The following diagram shows the possible lifecycle of an overdue event:


Event Lifecycle
Canceled
Pending
Waiting
Trash
Complete
Key:
• Dashed Line – Done by a user
• Solid Line – Done by the system

Overdue events are initially created in the Pending state. An event can take myriad paths after it's created; it all depends on how you've configured the system. The following topics describe an event's lifecycle:

How and When Events Are Activated

An overdue event contains the date is should be activated; this is referred to as its trigger date. On this date, the Overdue / Cut Event Manager (a background process (C1-ODET)) invokes the Event Activation algorithm plugged-in on the event's event type. The Event Activation algorithm, in turn, will decide on the state in which to leave the overdue event (e.g., it may transition it to the Complete state or the Waiting state).

If a user can't wait for the Overdue / Cut Event Manager real-time, they can click a button on Overdue Process - Main to activate (and recursively trigger) events online / real-time.

You control how an event's trigger date is populated by configuring the overdue process template. You are given two choices when you link an event type to an overdue process template:

- You can indicate the event should be assigned a trigger date when it is first created. You'd use this approach on the first event and events with no dependencies on earlier events. The following points describe how to configure the overdue process template to do this:
  - Indicate the event type is NOT dependent on other events, and
  - Define the number of days after the process's creation to use when calculating the trigger date.
- You can indicate the event should be assigned a trigger date only after earlier events are Complete. This technique should be used whenever you have an event that is only executed after other events are Complete. The following points describe how to configure the overdue process template to do this:
  - Indicate the event is dependent on other events, and
  - Define the number of days after the completion / cancellation of all dependent event(s) that the trigger date should be set to. The Overdue / Cut Event Manager sets the trigger date on such an event when it detects that all of its dependent events are complete / canceled.

NOTE:
Calendar vs Workdays. When an overdue event is created by the system, its trigger date is set in accordance with your date arithmetic preferences. Refer to Calendar vs. Work Days for more information.

Activating Events Should Add a Log Entry

As described above, an overdue process has a log holding a history of meaningful events in the process's life. Most Event Activation algorithms will add an entry to the process's log.

These log entries are more than just an audit trail as they also reference the objects that are created during activation. For example, if an activation algorithm creates a customer contact, the ID of the customer contact will be referenced on the log (and end-users will be able to drill down on the log entry to see the customer contact).

Holding Events

You can prevent a pending event with a trigger date on / before the current date from activating by plugging-in a Hold Event Activation plug-in on the overdue process template. This might prove useful, for example, if you want to suspend an overdue process while a case used to investigate the assertion of a customer is outstanding. Then, when the case closes, the overdue process can start up where it left off.

This technique would prove useful, for example, if your users can grant ad hoc suspend periods (e.g., if a customer wants a few extra days to pay before a cutoff). To do this, create a case type that has two states: Open and Close. Make sure to set up the Open state to have an automatic transition algorithm to close the case after X days. Then, all a user has to do is create a case of this type when they want to provide a suspend period. When the suspend period is over and payment isn't received, the case will close and the overdue process will start up where it left off.

Some Events Wait For Something Before They Activate

Consider this scenario - you want an overdue event to create a To Do entry so a user can authorize the next phase of an overdue process. When this event activates, the event's activation algorithm will create a To Do entry, but it will NOT transition the event to complete. Rather, the overdue event will exist in the waiting state. While in the waiting state, the Overdue / Cut Event Manager will monitor the state of the To Do entry. When the To Do entry completes, the original overdue event can transition to the complete state and then latter dependent events can be triggered. The following points describe how to configure the system to support this type of event:

- The event type's Event Activation algorithm should behave as follows:
  - It creates the object on which the overdue event waits.
  - It must link this object to the overdue process by creating a log entry where the prime-key of the related object is referenced (in a foreign-key characteristic). This log entry should also reference the event.
  - It should leave the overdue event in the waiting state.
- The event type must have a Monitor Waiting Event algorithm. This algorithm is invoked each time the Overdue / Cut Event Manager executes. If the related object has transitioned to a "final" state, the originating overdue event is transitioned to the complete state (and then latter dependent events are triggered).

> **NOTE:**
> **Bottom line.** Two algorithms must be set up on an overdue event type to implement waiting functionality: an Event Activation algorithm that creates the monitored object and a Monitor Waiting Event algorithm to check on the state of the monitored object. The Overdue / Cut Event Manager has the dual responsibility of activating the event and monitoring its related object for completion (and then triggering the dependent events when it completes).

While the above example illustrated how an overdue event could create and then monitor a To Do entry, you can use this functionality to create and monitor any object that has an initial and final state. If the base package does not contain the algorithms you need, simply develop new ones using the base-package algorithms as examples.

How Are Events Canceled

A pending event will be cancelled automatically by the system when the overdue process is canceled. Refer to How Are Overdue Processes Cancelled for more information.

A user may cancel a pending or waiting event at their discretion.

Regardless of what triggers the cancellation, the Cancel Logic algorithm plugged in on the overdue event type handles the cancellation. This allows you to introduce additional cancellation logic should the need arise. Please note that the base package cancel algorithms insert a log entry when a user manually cancels an event.

The Big Picture Of Cut Processes

While not required, many overdue processes will contain one or more events that will cut (i.e., stop) service. When such an event activates, it creates a "cut process". A cut process is very similar to an overdue process; the major difference is that a cut process is linked to a specific service agreement (the one being cut), whereas an overdue process is linked to the object(s) in arrears.

The topics in this section provide background information about cut processes.

How Are Cut Processes Created?

The activation of an overdue event can create one or more cut processes.

In addition, a user can manually create a cut process at their discretion using Cut Process - Main.

Overdue Events Wait For The Cut Process To Conclude

Overdue events that create cut processes are perfect examples of events that wait for the object they create to complete before they, in turn, complete. After the cut process concludes, the originating overdue event will complete thus triggering its dependent events.

Cut Processes Exist Under An Overdue Process

It's important to note that a cut process exists in respect of an overdue process. In other words, you can't create a cut process without referencing a parent overdue process. There are two reasons why:

- A cut process's overdue process defines the objects in arrears. It is these objects that are monitored to determine if the cut process (and overdue process) can be cancelled.
- The log associated with a cut process's overdue process contains the history of the cut process(es) and their events; there is no log specific to a cut process.

The Components Of A Cut Process

Cut processes are simpler than overdue processes as they simply contain one or more "cut events". These events are the service agreement-specific actions designed to encourage the customer to pay. For example, you might set up cut events that:

- Create field activities to cut service
- Send letters (via the creation of a customer contact)
- Create To Do entries
- Impact the account's credit rating
- Create a case (e.g., to manage a customer with life support issues)
- ... (the list is only limited by your imagination as algorithms are used to perform the event's actions)

You define the number and type of events by configuring cut process templates. When the system creates a cut process, it copies the events defined on the related template.

It's important to note that all cut events are created when the cut process is created. The Overdue / Cut Event Manager is responsible for activating the cut events at the appropriate time. Activation of an event causes the system to do whatever the event indicates (e.g., send a letter, send a To Do entry to a user, create a field activity, etc.)

Cut processes do not have their own log as their history is maintained on the parent overdue process's log.

Cut Process Information Is Overridable

"Cut process info" is the concatenated string of information that summarizes a cut process throughout the system. The base-package logic constructs this string by concatenating the following information:

- Its cut process template's description
- Its status
- For active processes, the number of days since it was created. For inactive processes, the number of days since it was inactivated.
- For active processes, the unpaid amount of the objects being collected on the cut process's overdue process

If you'd prefer a different info string, you can develop a new algorithm and plug-it in on your cut process templates. This design allows some / all cut process templates to have an override info string.

How Are Cut Processes Cancelled?

A user may cancel a cut process at their discretion, online / real-time using Cut Process - Main.

The system will automatically cancel a cut process when the overdue process is cancelled.

Regardless of what triggers the cancellation, the Cancel Cut Process algorithm plugged in on the cut process's template handles the cancellation.

If a cut process is canceled and the cut process's events created field activities, the cancel logic embodied in the algorithm can be quite sophisticated. Refer to Field Activities To Cut And Reconnect Service for an example.

Cut Processes and Events Are Created From Templates

When a cut process is created, its cut events are created by copying a cut process template's event types. Cut events follow the same lifecycle and possess the same behavior as overdue events.

Cut Events Are Like Overdue Events

Cut events almost identical to overdue events. The major difference is that cut events exists under a cut process and therefore are oriented towards the cut process's service agreement. The following topics describe other differences between cut events and overdue events.

Field Activities To Cut and Reconnect Service

A cut process that's created for a service agreement whose service can be severed will typically contain a cut event that creates field activity(s) to cut service. The base-package cut event activation algorithm will create field activities for the service points linked to the service agreement being cut. The type of activity is defined on the field activity type profile defined on each service point's SP type. Please see C1-CE-CR-FA for the exact details of the base-package algorithm.

If the cut process is canceled after field activity(s) are created and before the cut process completes, different activities can transpire. The specifics of what happens are controlled by a Cancel Logic algorithm plugged in on the cut process template. The following points summarize how a base-package algorithm (C1-CP-DWFA) works:

- It assembles all Pending, Held and Complete cut for nonpayment field activities (FA) created by the cut process (these are defined in the overdue process's log).
- Each FA is processed as follows:
    - If the FA is Pending or Held and it is not linked to a field order (i.e., it hasn't been dispatched):
        - The FA is cancelled
        - An entry is added to the overdue process log to indicate the cancellation
    - If the FA is Pending or Held and it is linked to a field order (i.e., it has been dispatched):
        - A To Do entry is created (the To Do Type is defined on the algorithm)
    - If the FA is Complete and the Cut Process's SA is pending start or active (i.e., it hasn't been expired yet):
        - Reconnect field activity(s) are created for every service point that had a cut for non-payment field activity. The type of activity is defined on the field activity type profile defined on each service point's SP type.

A Cut Event Expires Service Agreements (and May Trigger Final Billing)

The last cut event is typically set up with an activation algorithm that expires the cut process's service agreement. If earlier cut events created "cut for non-payment" field activities, these field activities will be used as the basis for stopping service. Refer to Finalizing Pending Stops for how the system use the meter reads on these field activities as the "stop reads" on the service agreement. Note, you can see the field activities that are used to "cut" and "stop" service by viewing the Field Activities grid on Service Agreement - Service Point.

➤ **NOTE:**
Final billing. When the last service agreement linked to an account is expired, the system will schedule the account for billing (outside of its normal bill cycle schedule).

Write Offs Are Implemented Using Overdue Events

The system has been designed to allow overdue events on the original overdue process to write-off the objects being collected. Refer to Writing Off Bills for a case-study explaining how to do this.

Calendar vs. Work Days

When you set up your overdue and cut process templates, you supply information that controls how each event's trigger date is calculated. You have two options:

- You can say that an event's trigger date can only be populated after earlier, dependent events are complete. For example, the 2 nd event (cut service) is triggered 2 days after the 1 st event is complete (48 hour warning letter).
- You can say that an event's trigger date is populated when the process is first created. You simply define the number of days after the start of the process when each such event should be triggered. For example, the 2 nd event (send cutoff warning) can be triggered 7 days after the start of the process.

In addition to the above, an option defined on the Feature Configuration for Overdue Processing plays a part in the calculation of an event's trigger date:

- If you set the option to use calendar days, the trigger date of events will be set to the first workday on / after the calculated date. For example, if you indicate that the 2 nd event is triggered 7 days after the 1 st event, the system will add 7 days to the 1 st event's completion date. It then checks if this is a workday (and not a holiday); if so, this is the trigger date of the event; if not, it assigns the trigger date to the next workday.
- If you set the option to use workdays, the trigger date will be calculated by counting workdays. For example, if you indicate that the 2 nd event is triggered 7 days after the 1 st event, the system will count 7 workdays and set the trigger date accordingly.

NOTE:
Account's division controls the work calendar. The system uses the above information in conjunction with the work calendar associated with the account's division to determine workdays.

Bill-Oriented Collection - Advanced Topics

The topics in this section provide information on features designed to facilitate the collection of overdue bills.

Miscellaneous Bill-Oriented Collection Topics

The topics in this section provide background information about a variety of bill-oriented collection topics.

Highlights Of The Sample Overdue Monitor Rule Algorithm

A base-package Overdue Monitor Rule algorithm exists to support many different types of overdue rules (see C1-CB-CR-RAT). Keep in mind that multiple such algorithms can be plugged in on Collection Class Overdue Rules. When the Overdue Monitor analyzes an account, it calls these algorithms until there are no more to invoke (or until an algorithm tells it that there is nothing more to check). The following points describe its various options:

- **Different rules based on credit ratings.** You can set up algorithms to only be applied to accounts with a credit rating <= a given value. We anticipate such algorithms will be used in conjunction with other such algorithms to apply different rules based on the customer's credit rating.

- **Different rules based on an account or service agreement characteristic.** You can set up algorithms to only be applied if an account or one of its service agreements has a given characteristic type and value effective within the last X days. For example, you could set up an algorithm that would process an account if it had a broken payment arrangement service agreement within the last 365 days (broken payment arrangement service agreements are marked with a given char type and value). You could set up a different algorithm that treated strategic customers more leniently (given that the definition of "strategic" could be held in an account or service agreement characteristic).
- **Skipping a bill if it has a future postpone date.** You can set up algorithms to ignore an unpaid bill if it has a "postpone date" that's in the future. This date would be defined on the bill using an ad hoc characteristic. The characteristic type is defined as a parameter on the algorithm.
- **Special process for bills with disputed debt.** You can set up an algorithm to create a given type of overdue process if the bill has disputed debt. By "disputed debt", we mean a bill with a financial transaction that's linked to an unbalanced match event that's marked as Disputed.
- **Special process for bills with credit balances.** You can set up an algorithm to create a special overdue process if the bill's total charges are negative (i.e., a credit bill). Such bills should be very unusual as we strongly recommend that credit bills should have their credit balance applied to an overpayment service agreement during bill completion.
- **Different rules when all service agreements are inactive.** You can set up an algorithm that should only be applied if all service agreements are inactive. We anticipate this can be used to differ the type of overdue process as if an account does not have active service agreements; there is nothing to stop to encourage them to pay.
- **Different rules based on age and amount of a bill.** You can set up an algorithm to create different types of overdue processes based on the age and amount of the bills.

> **NOTE:**
> **You are not limited by the base-package's rules.** If your implementation requires options that are not supported by the base-package algorithm, simply develop your own.

Holding A Process's Events

Refer to Holding Events for a description of how to prevent the activation of an overdue / cut processes events while a given condition is true. The condition is defined in an algorithm so you can set up the system to prevent event activation for practically any condition. For example, you might want to hold event activation while the bill's account has an active case (e.g., a user might set up a case to conduct an ad hoc investigation of billing discrepancies). When the case closes, you'd want the process to start up where it left off.

Bill Info Shows Unpaid and Write-off Amounts

The base-package Bill Information algorithm (plugged in on the Installation record) is responsible for constructing a bill's summary information that appears throughout the system. This algorithm can be configured to show the unpaid and write-off amounts of each bill.

Writing Off Bills

The topics in this section provide background information on how to write off bills.

After Service Is Cut, Write-Off Oriented Events Can Commence

After the last service agreement has been stopped, overdue events to manage the write-off of the overdue process's bill(s) should be triggered. Please note that a separate overdue process is NOT created to manage write-offs. Rather, events associated with the original overdue process will handle the write-off activities.

Collection Agency Referrals

Before debt is written off, many implementations refer the unpaid bills to a collection agency. The following points describe how to implement this.

- Set up an overdue event type with an activation algorithm that refers an overdue process's bill(s) to a collection agency.
- When such an event is activated, the system creates a Collection Referral record for a collection agency. The specific agency to which the debt is referred is controlled by the event type's activation algorithm. The sample algorithm simply refers debt to the collection agency with the least amount of referred debt. If you prefer different logic, you must develop your own algorithm.
- A collection agency referral history record is linked to an account. It contains the amount of debt referred to the collection agency. It is the creation of this record that triggers the interface of information to the collection agency. The method used to interface the information to the agency is defined on the collection agency's record. Refer to Setting Up Collection Agencies for more information.
- If the collection agency is successful in obtaining the funds, a payment will be added. If the payment satisfies the cancel criteria defined on the overdue process template's cancellation plug-in, the overdue process will cancel. When an overdue process is cancelled, the cancel criteria on the overdue process's template are executed. We strongly recommend plugging in an algorithm that will cancel collection agency referrals when an overdue process is cancelled.
- If the collection agency is not successful in obtaining your funds after a given amount of time, you probably want to cancel the referral and write-off the debt. The cancellation of the referral will happen automatically if you set up your overdue process template to have an event that creates a collection agency cancellation X days after the referral. You can cancel a referral manually by simply creating a new collection agency referral history record (with a type of cancel).
- Collection agencies are notified of the cancellation of a referral by the creation of a new collection agency referral history record (with a type of cancel). This record will be interfaced to the agency in the same manner used to interface a new referral (see above).

➤ **NOTE:**
Log entry. The base-package overdue event activation algorithms that make and cancel collection agency referrals insert rows in the overdue process's log to audit these events.

Writing Off Bills Will Create Write-Off Adjustments And Possibly A Match Event

Most overdue process templates will be configured to contain an event that writes-off the unpaid balance of bills.

➤ **NOTE:**
Processing is redirected to another algorithm. The base-package overdue event activation algorithm that writes off bills simply redirects the call to the Write-Off Bill algorithm plugged in on the account's Collection

> Class Overdue Rules. The reason for this redirection is because users can manually write-off a bill (using Bill - Main) and when they do this, we want to invoke the same logic as when an overdue process writes-off a bill.

The base-package Collection Class Overdue Rules - Write-Off Bill algorithm determines the amount that should be written off for each distribution code on each unpaid financial transaction. It then creates a separate adjustment for each service agreement where the lines on the adjustment contain the amount to be written off for each distribution code. If the unpaid financial transactions are not linked to a match event, the write-off adjustments plus the unpaid financial transactions will be linked to a new match event. If the unpaid financial transactions were linked to an unbalanced match event, the write-off adjustments will be added to the existing match event (thus making it balanced).

If partial payments were made against a bill, the amount written off will be prorated in light of the partial payments. For example, if 20% of a bill had been paid, 80% of each distribution code will be written off.

For example, assume the original bill's FT looked as follows (note, if the bill has multiple service agreements this will need to be done for each SA's FT's):

- Debit A/R $110.00
- Credit Flat Charge Revenue ($50)
- Credit Usage Revenue ($50)
- Credit City Tax Payable ($5)
- Credit State Tax Payable ($5)

Assume the customer had made a partial payment of $11 and it was matched to this FT. At write-off time, the system will create an adjustment whose adjustment lines will cause each distribution code to be written off by 90% (100% - $11 / $110). For example:

- Debit Flat Charge Revenue (or some W/O Expense) $45
- Debit Usage Revenue $45
- Debit City Tax Payable $4.50
- Debit State Tax Payable $4.50
- Credit A/R $99

This adjustment will then be linked to the original match event on which the payment was linked (thus making it Balanced).

> ➤ **NOTE:**
> **Log entry.** The base-package overdue event activation algorithm that writes off bills inserts a row into the overdue process's log for each bill written off.

Small Amount Write-Downs

Many organizations will write-down a bill whose value is small early in an overdue process. The base-package overdue event activation algorithm has parameters to support this requirement (this algorithm allows you to write-off an overdue process's bill(s) if their value is less than a threshold amount).

If your organization writes-down small amounts differently than large amount, simply set up an overdue event type to reference such an activation algorithm and position it in the appropriate place in the overdue process template.

Bill Info Shows Written Off Amounts

The base-package Bill Information algorithm (plugged in on the Installation record) is responsible for constructing a bill's summary information that appears throughout the system. This algorithm can be configured to show the amount written-off for each bill.

Online Write Off Of Bills Is Performed On Bill-Main

If a bill's account is associated with a Collection Class Overdue Rule that has a Write Off Bill algorithm, a button appears on Bill - Main if the bill hasn't been written off. When clicked, the Write Off Bill algorithm is invoked to write-off the bill.

Online Reversal Of Written Off Bills Is Performed On Bill-Main

If a bill's account is associated with a Collection Class Overdue Rule that has a Write Off Bill algorithm, a button appears on Bill - Main if the bill has been written off. When clicked, the Write Off Bill algorithm is invoked (in reversal mode). The algorithm cancels the write-off adjustments (thus making the bill payable again). It also schedules the account for review by the Overdue Monitor the next time it runs.

Write Offs And Overdue Process Cancellation

It should be stressed that writing-off a bill may cause an overdue process to be canceled because the bill's FT will be linked to a balanced match event (note, the specific cancel criteria are in a plug-in so this is not a hard rule).

The following points highlight interesting aspects of bill write-off and overdue process cancellation:

- If a user writes off a bill real time and the bill has an Active overdue process, the process's cancel criteria will be invoked. This typically results in the cancellation of the overdue process.
- If an overdue event writes off a bill, the state of the process depends on your cancel criteria and where the overdue event is positioned in the overdue process. For example,
  - Imagine an overdue process that has an overdue event that writes offs small amounts of debt early in the process. If such a process is applied against bill with a small amount of debt, the process will be canceled (when the event activates).
  - Contrast this to an overdue process where the last event writes off the bill. Because there are no other events to activate, the process will complete (i.e., it will not be canceled).

An Alert Can Show Written Off Bills

A base-package Control Central Alert algorithm (C1-WO-BILL) can be set up to highlight if the account in context has written off bills. This algorithm is plugged in on the Installation record.

Bill-Oriented Payment Arrangements

A payment arrangement is an agreement with a customer to payoff severely overdue debt in **billed** installments. Bills sent to customers with payment arrangements contain charges for both their current services and their payment arrangement installment amount.

➤ **NOTE:**

Nomenclature. Some organizations refer to payment arrangements as "current bill plus" agreements because the customer's bills contain charges for both their current debt plus their installment amount. After the customer has paid off their overdue debt, the payment arrangement closes and the customer's bills only contain charges for their current debt.

The topics in this section describe how to set up a payment arrangement and how the system monitors the ongoing arrangements.

Creating Payment Arrangements

When you create a payment arrangement, you are actually creating a service agreement. This service agreement is just like other service agreements in that:

- It holds debt.
- It is periodically billed (thus creating unmatched bill segment financial transactions).
- When a payment is received, the payment segment financial transactions are matched to the bill segment financial transactions.

Debt is transferred to a payment arrangement service agreement (PA SA) from the customer's delinquent bills at the inception of the payment arrangement.

When you transfer debt from the overdue bills to a PA SA, transfer adjustments are created to transfer debt from the delinquent SAs to the PA SA. Match events are created to link the "transfer from" adjustments to the original unpaid financial transactions (FT). When all FT's on a bill are linked to balanced match events, the bill is no longer considered overdue and any active overdue process (and its related cut processes) will be cancelled. Refer to How Are Overdue Processes Canceled for more information.

➤ **NOTE:**

Use the Payment Arrangement Transaction. Use the Payment Arrangement - Bill Oriented to set up bill-oriented payment arrangements. This transaction creates a PA SA, transfers overdue bills to it, and sets up the installment amount. This transaction is also used if you need to break or cancel the payment arrangement.

Installment, Payoff and Current Amounts

WARNING:

If you do not understand the difference between payoff balance and current balance, refer to Current Amount versus Payoff Amount.

When you set up a payment arrangement service agreement (PA SA), you transfer delinquent debt to the PA SA using transfer adjustments. After moneys are transferred, the system sets the PA SA's current balance to zero. At this point, there will be no overdue bills. If the customer neglects to pay bills containing charges associated with the payment arrangement, an overdue process will ensue.

PA SAs start their life with a non-zero payoff balance (i.e., they have debt when first started). This debt is transferred from the bills whose outstanding debt necessitated the creation of the PA SA.

The installment amount that the customer is billed is determined by the number of installments used to payoff the debt. For example, if the customer owes $500 and they want to pay this off in 10 installments, you'd set up the installment amount to be $50. The installment amount is saved on the PA SA's recurring charge amount. If the customer again falls into arrears on their bills, you can transfer additional bills to the PA SA. You can also change the installment amount as needed.

A PA SA's payoff balance typically differs from its current balance. The payoff balance is the amount of debt remaining to be paid off under the terms of the payment arrangement. The current balance is the installment amount that has been billed but not paid. For example, a customer who is paying off $500 with 10 installments of $50 would have an initial payoff balance of $500 and a current balance of $0. After the first bill, the PA SA would still have a payoff balance of $500, but its current balance would be $50. When the customer pays, the PA SA's payoff balance would fall to $450 and its current balance would return to $0.

The following table contains a financial example of a customer who sets up a payment arrangement to payoff $1,000 of debt in $10 installments.

Event	Normal SA's GL Accounting	PA SA's GL Accounting	Normal SA's Current Balance	Normal SA's Payoff Balance	PA SA's Current Balance	PA SA's Payoff Balance
Prior to creation of payment arrangement	N/A	N/A	1000	1000	N/A	N/A
Transfer debt from normal SA(s) to PA SA	Xfer 1000 A/R <1000>	PA A/R 1000 Xfer <1000>	0	0	1000	1000
Set current balance to zero on PA SA	N/A	N/A	0	0	0	1000
Customer is billed ($50 for new debt and $10 of payment arrangement debt)	A/R 50 Revenue <50>	N/A	50	50	10	1000
Customer pays $60	Cash 50 A/R <50>	Cash 10 PA A/R <10>	0	0	0	990

When the customer pays off the payment arrangement debt, the system automatically closes the PA SA after it final bills (assuming the PA SA's SA type references a bill segment type that has a bill segment creation algorithm of Recurring Charge With Auto Stop).

Breaking A Bill-Oriented Payment Arrangement

If a customer neglects to pay a bill, an overdue process is created. If the bill contains charges from a payment arrangement, you'll want to "break" the payment arrangement. By "break", we mean cancel the arrangement and reinstate the debt under the originating bills. To do this, you must configure the cut process template used to cut the PA SA to contain a cut event that breaks the payment arrangement.

➤ **NOTE:**
Processing is redirected to another algorithm. The base-package cut event activation algorithm that breaks a payment arrangement simply redirects the call to the Bill-Based Payment Arrangement algorithm plugged in on the account's Collection Class Overdue Rules. The reason for this redirection is that users can manually break a payment arrangement (using Payment Arrangement - Bill Oriented). When they do this, we want to invoke the same logic as when a cut event activation algorithm breaks a payment arrangement.

The base-package Collection Class Overdue Rule - Bill-Based Payment Arrangement algorithm does the following:

- Creates "cancel adjustments" for all unpaid financial transactions associated with the payment arrangement (e.g., billed, but not paid installments).
- Cancels ALL adjustments that were used to transfer the debt to the payment arrangement. When these are cancelled, the original bills will become unpaid (and the debt will be rather old by this point).
- Syncs up current balance with payoff balance on the PA SA.
- Makes the PA SA pending stop and expires it (SA activation will stop the SA when it next runs).
- If there is a credit left on the PA SA (because payments were made against the arrangement), the credit will be distributed amongst the bills that contributed debt to it (the oldest bills are paid off first). This relief will occur via transfer adjustments from the PA SA to the original SAs.
- Note, if there is a debit left (e.g., because LPC were issued or some other type of adjustment was created by an operator), an error is issued as we don't handle new debit on a payment arrangement.
- Inserts a characteristic on the PA SA to indicate that it has been broken (you might want to use this to cause more strict overdue rules to be applied to accounts with broken payment arrangements).
- Marks the account so it will be reviewed by the Overdue Monitor the next time it runs.

➤ **NOTE:**
The PA SA must final bill before it closes. It's important to note that the PA SA will only close after the PA SA is final billed. This is OK as it won't have any money left on it.

When the Overdue Monitor next runs, it will analyze the account's reinstated bills. We recommend setting up a Collection Class Overdue Rule - Overdue Monitor Rule algorithm that has more stringent rules if the account has a broken payment arrangement within the last X days. The base-package algorithm (C1-CB-CR-RAT) supports this type of processing.

➤ **NOTE:**
Log entry. The base-package overdue event activation algorithm that breaks a payment arrangement SA inserts a row into the overdue process's log.

Online Breaking

A user can click a button on the Payment Arrangement - Bill Oriented page to break a payment arrangement real time. When clicked, the Collection Class Overdue Rule - Bill-Based Payment Arrangement algorithm is called to break the payment arrangement. This is the same algorithm called when a cut event break a payment arrangement.

Online Canceling

A user can click a button on the Payment Arrangement - Bill Oriented page to cancel a payment arrangement real time. Cancellation should be used when you want to "logically delete" a PA SA because it shouldn't have been created.

The logic described above for breaking a payment arrangement is executed when a user cancels a payment arrangement; the only difference is that the PA SA is marked with a different characteristic type / value than when it is broken. The "broken" characteristic type / value can be used to apply stricter rules to the account when it's next reviewed by the Overdue Monitor.

Creating Overdue and Cut Procedures

Your overdue procedures define how your organization collects overdue debt. Your cut procedures define how your organization cuts (i.e., stops) service agreements when collection attempts fail. In this section, we describe how to set up the data that controls these procedures.

> **WARNING:**
> There are numerous ways to design your overdue and cut procedures. Some designs will result in easy long-term maintenance; others will result in maintenance headaches. In this section, we provide information to help you understand the ramifications of the various options. Before you set up your overdue and cut procedures, we encourage you to gain an intuitive understanding of these options by using the system to prototype the alternatives.

Designing Your Overdue Procedures

The design of your overdue procedures is an iterative process. Over time, you will develop skills that will allow you to skip some steps. However, when you're starting out, we recommend you use the following matrix as your guide. When the matrix is complete, you're ready to set up the overdue processing control tables.

Account's Division	Account's Collection Class:	Account's Collection Class:

The topics discussed below will gradually complete this matrix using a simple case study.

- **FASTPATH:**
  For more information about how the information in this matrix is used to monitor your customers' debt, refer to How Does The Overdue Monitor Work.

Your Divisions Are Frequently Preordained

An account's division defines the jurisdiction whose rules govern the account. For example, an account's division controls how its payments are distributed, if / how late payment charges are levied, etc. Divisions have typically been designed in advance of designing your overdue rules.

In our example, we assume you have two divisions: North and South.

Account's Division	Account's Collection Class:	Account's Collection Class:
North		
South		

Designing Your Collection Classes

Multiple collection classes are needed when your organization has different overdue rules and / or procedures based on the type of customer. If all customers are treated the same way, you'll have a single collection class (call it Generic). However, if you're like many organizations, you will have multiple collection classes.

For example, for commercial/industrial customers, you probably don't worry until they owe you more than, say, $100 after 20 days. For residential customers, you probably don't worry until they owe you more than, say, $5 after 20 days. In this situation, you will have at least two collection classes: one for commercial/industrial customers, the other for residential customers.

In our example, we assume you have two collection classes: Residential and Commercial/Industrial.

Account's Division	Account's Collection Class:	Account's Collection Class:
	Residential	Commercial/Industrial
North		
South		

NOTE:
There are two very different ways to monitor your accounts for overdue debt. This chapter describes the method referred to as Overdue Processing. Refer to Collection / Severance / Write Off for a description of the other method. We anticipate that most organizations will only use a single method. If your organization opts to use both methods, you will need to set up the corresponding collection classes.

Designing Overdue Monitoring Rules

At this point, we have the rows and columns defined in our matrix. Now it's time to work on the individual cells.

NOTE:
Single currency. We've assumed that your implementation works in a single currency. If this is not true, you will need to add a 3rd dimension that will have a value for each currency code.

Each cell will contain the rules that the Overdue Monitor uses to determine if an account has overdue debt. These rules will eventually be configured using one or more algorithms on Collection Class Overdue Rules.

Account's Division	Account's Collection Class:	Account's Collection Class:
	Residential	Commercial/Industrial
North		
South		

NOTE:
If the Overdue Monitor encounters an account whose collection class and division does not have overdue rules set up, it will issue an error.

Determining the rules in each cell can be straightforward or complicated; it depends on how your organization works. Our case study assumes the following:

- For residential debt (regardless of division) we have the following rules:
  - Highest priority. If a bill exists with unpaid FT's > $0 that is older than 50 days, create the "accelerated overdue process" for residential customers. We'll talk more about this process later.
  - Medium priority. If the account's has a broken payment arrangement within the last 60 days with an unpaid bill > $0 that is older than 20 days, create the "broken payment arrangement overdue process".
  - Lower priority. If a bill exists with unpaid FT's > $25 that is older than 25 days, create the "courtesy reminder overdue process" for residential customers. We'll talk more about this overdue process later.
- For commercial-industrial debt (regardless of division) we have the following rules:
  - Highest priority. If a bill exists with unpaid FT's > $0 that is older than 45 days, create the "commercial 45 days late overdue process". We'll talk more about this process later.
  - Medium priority. If the account's credit rating is < 550 and has an unpaid bill > $0 that is older than 20 days, create the "risky commercial customer overdue process".
  - Lower priority. If a bill exists with unpaid FT's > $100 that is older than 30 days, create the "commercial 30 days late overdue process". We'll talk more about this overdue process later.

Given the above, our matrix will look as follows:

Account's Division	Account's Collection Class:	Account's Collection Class:
	Residential	Commercial/Industrial
North	Highest Priority: If a bill exists with unpaid FT's > $0 that is older than 50 days, create the accelerated overdue process for residential customers. Medium Priority: If the account has a broken payment arrangement within the last 60 days with an unpaid bill > $0 that is older than 20 days, create the broken payment arrangement overdue process. Lowest Priority: If a bill exists with unpaid FT's > $25 that is older than 25 days, create the courtesy reminder overdue process for residential customers.	Highest Priority: If a bill exists with unpaid FT's > $0 that is older than 45 days, create the commercial 45 days late overdue process. Medium priority. If the account's credit rating is < 550 and has an unpaid bill > $0 that is older than 20 days, create the risky commercial customer overdue process. Lowest Priority: If a bill exists with unpaid FT's > $100 that is older than 30 days, create the commercial 30 days late overdue process.

South	Same as above	Same as above

NOTE:

The rules are limited by your imagination (and business requirements). While the base-package Overdue Monitor Rule algorithm (C1-CB-CR-RAT) supports the above scenarios, we'd like to stress these are just examples. Your implementation can operate using very different rules by either configuring the base-package algorithm (it has many parameters that you can use to tailor your rules) OR by introducing a new algorithm. Refer to Highlights Of The Sample Overdue Monitor Rule Algorithm for the options delivered with the base-package.

Designing Overdue Process Templates and Event Types

The following table shows a sample overdue process template for one of the rules in the Residential / North cell in the previous section's matrix.

Overdue Process Template	Overdue Event Type	When Triggered
Accelerated overdue process for residential customers	Old debt letter	At inception of process
	Cut active service agreements	10 days after inception
	Reduce customer's credit rating	10 days after inception (i.e., at the same time the cut process is created)
	Write down small debt	0 days after completion of the cut process(es)
	Refer debt to collection agent	0 days after attempting the small write down (this means that either the small write-down or the agency referral will take place as if the write-down is successful, the bill's FTs will be matched to balanced match events and the overdue process will stop)
	Cancel collection agent referral	45 days after referral
	Write-off debt	0 days after collection agent cancellation

You should create a similar table for each of the distinct overdue process templates in your matrix.

At this point, you've designed the distinct overdue process templates. Next, you'll need to design the algorithms that control their overdue processes:

- A template's Calculate Unpaid and Original Amount algorithm calculates the original and unpaid amounts of the objects being collected by a process. These values are used throughout the overdue processing module.
- A template's Cancel Criteria algorithm is executed to determine if a process should be cancelled. Refer to How Are Overdue Processes Cancelled for the details.
- A template's Cancel Logic algorithm is executed to cancel a process. Refer to How Are Overdue Processes Cancelled for the details. Please note that the logic embodied in this type of algorithm can be sophisticated because it is responsible for stopping an ongoing process's activities (e.g., this could involve cancelling field activities or cases). Cancellation algorithms are also responsible for inserting log entry(s).
- A template's Hold Event Activation algorithm is invoked to determine if the Overdue / Cut Event Manager should suspend the activation of the process's events.

- A template's Information algorithm is invoked to construct the override "info string".

Next, extract each unique event type from the above table:

Overdue Event Type	Action
Old debt letter	Create a customer contact
Cut active service agreement(s)	Start a cut process for every active SA with an unpaid FT on the bill
Reduce customer's credit rating	Insert an account credit rating history record
Write down small debt	Create write-down adjustments if unpaid debt is less than $x
Refer debt to collection agent	Create a collection agency referral
Cancel collection agent referral	Cancel the collection agency referral
Write off unpaid debt	Create adjustments to write-off unpaid debt

At this point, you know the distinct event types. Next, you'll need to design the algorithms that control the lifecycle of each event type:

- The event type's Event Activation algorithm(s) are executed by the Overdue / Cut Event Manager on its trigger date. The following points describe the logic embodied in such an algorithm:
  - The activity that happens on the trigger date (e.g., creation of a customer contact, To Do, etc.). Refer to Overdue Events Can Do Many Things for the details.
  - Whether the event is transitioned into the Waiting or Complete state when it's triggered. Refer to Some Events Can Wait for the details.
  - How the log entry(s) associated with event activation will be constructed. The base-package algorithms allow you to control the verbiage in the log entry by defining the desired message number on the algorithm. This means that you may have to set up new messages. Refer to Activating Events Should Add A Log Entry for the details.
- The event type's Cancel Logic algorithm(s) are invoked when an event is cancelled. The following points describe the logic embodied in such an algorithm:
  - If the event is allowed to be canceled. This logic may be necessary if some conditions prevent events of this type from canceling. For example, you may want to prevent an event from canceling when there are later dependent events that aren't canceled.
  - Any ancillary actions that take place during cancellation.
  - How the log entry(s) associated with event cancellation will be constructed.
- The event type's Monitor Waiting Event algorithm(s) are invoked to monitor a waiting event. These algorithms are responsible for transitioning a Waiting event to Complete if the object on which it's waiting is complete.
- The event type's Event Information algorithm is invoked to construct the override "info string".

Once you've designed each event type's algorithms, you're ready to design your cut processes.

Designing Cut Process Templates and Event Types

While not required, many overdue processes will contain an overdue event that cuts (i.e., stops) service (in the hopes that stopping service will inspire the customer to pay). When such an event activates, it creates one or more cut process(es).

The system allows you to control if and how service is cut by setting up Cut Process Rule algorithms on your SA types. This allows different rules for different types of service agreements. In addition, because these rules are embodied in algorithms, a given SA type can have conditional logic that controls the type of cut process created. For example, you may have a different cut process if the customer has life support equipment, or if it's winter, or ...

NOTE:
If a service agreements of a given type should never be cut. If you have certain SA types that should never be cut, do NOT define an algorithm in the SA type's Cut Process Rule .

Determining your Cut Process Rules can be straightforward or complicated; it depends on how your organization works. Our case study assumes the following:

- For service agreements associated with metered service, we have the following rules:
  - Highest priority. If the customer has life support requirements, create "cut process in light of life support" process.
  - Lowest priority. Otherwise, create a "metered service" cut process.
- For service agreements associated with charitable contributions, create an "expiration only" cut process.

Once you've determined if / how each SA type is cut, you need to design your cut process templates (you'll need one for each unique method of cutting a service agreement). The following table shows the cut process templates referenced above. Adjacent to each process are its events and an indication of when they are triggered.

Cut Process Template	Event Number	Cut Event Type	Dependent On Event(s)	Trigger Date Set To X Days After Completion Of Dependent Events
Metered service	10	Letter - 48 hour disconnect for non-payment warning	N/A - first event	0
	20	Field activity - disconnect for non-payment	10	2
	30	'Service has been disconnected' letter	20	0
	40	Expire service agreement	20	10
Cut process in light of life support	10	Generate delinquent life support customer To Do entry (seeking approval for the cut)	N/A - first event	0
	20	Letter - 72 hour disconnect for non-payment warning	10	0
	30	Generate impending life support cutoff To Do entry to C&C rep	20	3
	40	Field activity - cut for non-payment	30	0
	50	Service has been disconnected letter	40	0
	60	Expire service agreement	40	10
Expiration only	10	Expire service agreement	N/A - first event	

At this point, you've designed the distinct cut process templates. Next, you'll need to design the algorithms that control the lifecycle of their cut processes:

- A template's Cancel Logic algorithm is executed to cancel a process. In addition to cancelling the event, these algorithms are also responsible for inserting log entry(s).
- A template's Information algorithm is invoked to construct the override "info string".

Next, extract each unique event type from the above table:

Cut Event Type	Action
48-hour warning letter	Create a customer contact - 48-hour warning letter (letters are created via customer contacts)
72-hour warning letter	Create a customer contact - 72-hour warning letter
Disconnect for non payment	Create a cut for non-payment field activity
Permission to start a cut process for a life support customer	Create To Do seeking permission to send 72 hour letter
Permission to create a cut field activity for a life support cutoff	Create To Do seeking permission to issue a cut field activity
Service has been disconnected letter	Create a customer contact - service cut letter
Expire service agreement	Expire service agreement

At this point, you know the distinct event types. Next, you'll need to design the algorithms that control the lifecycle of their events:

- The event type's Event Activation algorithm(s) are executed by the Overdue / Cut Event Manager on its trigger date. The following points describe the logic embodied in such an algorithm:
  - The activity that happens on the trigger date (e.g., creation of a customer contact, To Do, etc.). Refer to Cut Events Can Do Many Things for the details.
  - Whether the event is transitioned into the Waiting or Complete state when it's triggered. Refer to Some Events Can Wait for the details.
  - How the log entry(s) associated with event activation will be constructed. The base-package algorithms allow you to control the verbiage in the log entry by defining the desired message number on the algorithm. This means that you may have to set up new messages. Refer to Activating Events Should Add A Log Entry for the details.
- The event type's Cancel Logic algorithm(s) are invoked when an event is cancelled. The following points describe the logic embodied in such an algorithm:
  - If the event is allowed to be canceled. This logic may be necessary if some conditions prevent events of this type from canceling. For example, you may want to prevent an event from canceling when there are later dependent events that aren't canceled.
  - Any ancillary actions that take place during cancellation.
  - How the log entry(s) associated with event cancellation will be constructed.
- The event type's Monitor Waiting Event algorithm(s) are invoked to monitor a waiting event. These algorithms are responsible for transitioning a Waiting event to Complete if the object on which it's waiting is complete.
- The event type's Event Information algorithm is invoked to construct the override "info string".

Set Up Tasks

The above topics provided background information about how overdue processing works. The following discussion summarizes the various set up tasks.

Cut Event Types

You will find that most of the time spent setting up your cut event types is spent setting up the objects that are referenced on the cut event type algorithms. For example, if you use the base-package algorithms, you may need to set up the following:

- The various "types" for the objects created by the plug-ins. For example, if a cut event type creates a To Do entry, you must supply the desired To Do type.
- Foreign key characteristic types that are used to reference the ancillary objects in the log entries (e.g., if an event creates a customer contact, the log references this customer contact using a FK characteristic type). Note, many of these will exist in the base-package.
- Messages that are used to define the verbiage in the log entries. For example, if you use the base-package algorithm that creates a customer contact, you must supply the desired message category and number that contains the verbiage that appears in the log when customer contacts are created. Note, messages have been set up for all base-package algorithms (this means you should not have to set up new messages).
- Etc.

The only way to compile the complete list is to design the parameters for each cut event type algorithm. Refer to Cut Event Type - Main for the supported plug-in spots.

After you've set up the objects referenced on the algorithms, you can then set up the algorithms. Only then can you set up the cut event types.

Cut Process Templates

After your cut event types exist, you can set up your cut process templates. You will find that most of the time spent setting up your cut process templates is spent setting up the objects that are referenced on the cut process template algorithms. Refer to Cut Process Template - Main for the supported system events.

SA Type - Cut Process Rules

After you've created your cut process templates, you can set up the algorithms that hold your cut process rules. These are plugged-in on SA Type - Algorithm in the Cut Process Rule system event.

Overdue Event Types

You will find that most of the time spent setting up your overdue event types is spent setting up the objects that are referenced on the overdue event type algorithms. For example, if you use the base-package algorithms, you will set up the following:

- The various "types" for the objects created by the plug-ins. For example,
  - If an overdue event type creates a To Do entry, you must set up the To Do type.
  - If an overdue event type creates a customer contact, you must set up the customer contact type.
  - If an overdue event type writes off debt, you must set up the adjustment types.

  - Etc.
- Foreign key characteristic types that are used to reference the ancillary objects in the log entries (e.g., if an event creates a customer contact, the log references this customer contact using a FK characteristic type). Note, many of these will exist in the base-package.
- Messages that are used to define the verbiage in the log entries. For example, if you use the base-package algorithm that creates a customer contact, you must supply the desired message category and number that contains the verbiage that appears in the log when customer contacts are created. Note, messages have been set up for all base-package algorithms (this means you should not have to set up new messages).
- Etc.

The only way to compile the complete list is to design the parameters for each overdue event type algorithm. Refer to Overdue Event Type - Main for the supported plug-in spots.

After you've set up the objects referenced on the algorithms, you can then set up the algorithms. Only then can you set up the overdue event types.

Overdue Process Templates

After your overdue event types exist, you can set up your overdue process templates. You will find that most of the time spent setting up your overdue process templates is spent setting up the objects that are referenced on the overdue process template algorithms. Refer to Overdue Process Template - Main for the supported system events.

Collection Classes

Set up collection classes as per your overdue procedures. Make sure to indicate that these collection classes use the Overdue collection method (only accounts linked to collection classes designated as using the Overdue collection method or processed by the Overdue Monitor).

Collection Class Overdue Monitor Rules

After your overdue process templates exist, you can set up your Overdue Monitor Rules. These rules are algorithms plugged in on Collection Class Overdue Rules. You will find most of the time spent setting up these algorithms is spent setting up the objects referenced on the base-package algorithm.

Feature Configuration

You must set up a Feature Configuration to define parameters that control various overdue processing options.

The following points describe the various **Option Types** that must be defined:

- Trigger Date: Y-Workdays, N-Calendar Days. This option controls how the system computes the trigger dates on overdue and cut events. Enter Y if the system should use workdays. Enter N if the system should use calendar days. Refer to Calendar vs Work Days for the details.
- Payment Arrangement Type (B/S/A). This option indicates whether your implementation uses the Balance-Oriented Payment Arrangements (value **S**), Bill-oriented Payment Arrangements (value **B**) or both (value **A**). The value governs the navigation path for the payment arrangement lines in the Account History and Credit & Collection zones.

- Champion Template$Challenger Template$Percentage(1-100). You need only set up options of this type if your implementation implements Champion / Challenger functionality. Options of this type are entered in the format A $B$nnn where A is the overdue process template of the champion template, B is the overdue process template of the challenger template, and C is the percent of the time that the system should create the challenger template. The overdue monitor uses this option to override the champion overdue process template X% of the time with the challenger template. You may enter any number of these options (but only one per Champion Template).

Overdue and Cut Event Cancellation Reasons

Overdue events can be cancelled automatically and manually (at the discretion of a user). Regardless of the method of cancellation, a cancellation reason must be supplied. You set up your overdue event cancellation reasons using Overdue Event Cancellation Reason - Main and Cut Event Cancellation Reason - Main.

Collection Agencies

If you refer debt to collection agents, you must set up your collection agencies.

Alert To Highlight Active Overdue Processes

If you want an alert to appear if the account has active overdue processes, you must configure an appropriate Control Central Alert algorithm (C1-OD-PROC). This algorithm is plugged in on the Installation record.

Bill-Oriented Collection - Additional Set Up

The topics in this section provide information on additional set up requirements if you collect on unpaid bills.

One Bill Per Match Event

As mentioned earlier, a bill is considered paid if its financial transactions (FTs) are linked to a balanced match event. To determine a bill's outstanding amount, FTs from different bills cannot be commingled on the same match event (but it's OK for a bill's FTs to be on multiple match events). If you stick by the rule of "just one bill per match event" you will then be able to determine the outstanding balance of a partially paid bill. However, if you mix more than one bill under a match event, a particular bill's balance may become indeterminate.

The following Open-Item algorithm types have been provided by the base package to help enforce this rule:

- The Distribute by Bill Due Date Payment Distribution algorithm (C1-PYDS-BDU).
- The match by Bill ID Payment Distribution Override algorithm (C1-PDOV-PYBL).
- The FT cancellation FT Freeze algorithm (C1-CFTZ-COFT).

If any of your customized plug-ins and processes create match events, it is important that these too enforce this rule. You may want to refer to the base package algorithms as an example of how to do this.

Bill-Based Payment Arrangements

If you set up payment arrangements for unpaid bills, you must configure the Bill-Based Payment Arrangement algorithm and plug it in on your Collection Class Overdue Rules.

It's important to set up the adjustment types used during payment arrangement creation to NOT print on bills. This is because the base-package algorithm will match the adjustments used to transfer debt to the payment arrangement with the adjustment used to reduce the payment arrangement's current amount by the amount of the transfer if all adjustment types are set up to not print.

Bill-Based Write-off

If you write-off unpaid bills, you must set up the following:

- Set up the adjustment type that will be used to write-off an unpaid financial transaction. This adjustment type must be configured as follows:
  - **Adjustment Amount Type** must be Calculated Amount
  - Its distribution code is irrelevant as a separate calculation line will be created for each distribution code on the FT's that is written off and these lines will reference the appropriate distribution code.
  - It must reference an adjustment type char type / value that identifies it as one used to write-off a bill's FTs
  - The following algorithms must be defined on the adjustment type:
- The Generate Adjustment system event must reference an algorithm that has the responsibility of determining how to write-off a FT. This algorithm should be determining the FT's GL details and creating a separate adjustment calculation line for each GL detail. The base package is supplied with a sample algorithm that does this (C1-ADJG-WO).
- The Adjustment Financial Transaction system event should reference an algorithm that impacts current, payoff and the GL by the amount being written off.
  - The distribution codes referenced on the financial transactions must be set up with a characteristic that holds the distribution code used to write-off the original amounts. For example,
    - Distribution codes used to record tax liabilities will typically reference the same distribution code for write-off (most organizations reverse tax liabilities at write-off time)
    - Distribution codes used to record revenue will typically reference a write-off distribution code used to record a write-off expense.
    - Distribution codes used to record receivables will typically not reference a write-off distribution code because receivables are implicitly written off when revenue and tax liabilities are written off.
  - Set up an adjustment cancellation code used when a users reverses a written-off bill (reversal involves canceling the write off adjustments).
  - Set up a Write Off Bill algorithm and plug it in on your Collection Class Overdue Rules. This algorithm will reference the adjustment type described in the previous point.

Alert To Highlight Written Off Bills

If you want an alert to appear if the account has bills with written-off debt, you must configure an appropriate Control Central Alert algorithm (C1-WO-BILL). This algorithm is plugged in on the Installation record.

Open-Item Bill Amount Plug-In

You must set up the algorithm that computes the original, unpaid, and write-off amounts of your open-item bills. This algorithm is called by other algorithms when these amounts are needed. This algorithm is plugged-in on Installation in the Determine Open Item Bill Amounts spot.

Setting Up Overdue Processing

The topics in this section describe how to set up the control tables to implement your overdue processing.

Setting Up Overdue Event Types

An overdue event type encapsulates the business rules that govern a given type of overdue event. Open **Admin** > **Credit & Collection** > **Overdue Event Type** > **Add** to set up overdue event types.

NOTE:
Recommendation. Before using this transaction, we strongly recommend that you review The Big Picture Of Overdue Events.

Description of Page

Enter a unique **Overdue Event Type** code and **Description** for the overdue event type.

Use **Long Description** to provide a more detailed explanation of the purpose of the overdue event type.

The **Algorithms** grid contains algorithms that control important functions. You must define the following for each algorithm:

- Specify the algorithm's **System Event** (see the following table for a description of all possible events).
- Specify the **Algorithm** to be executed when the System Event executes. 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.

The following table describes each **System Event** (note, all system event's are optional and you can define an unlimited number of algorithms for each event).

System Event	Optional / Required	Description
Cancel Logic	Required	This algorithm is executed to cancel an overdue event. Refer to How Are Events Canceled for the details. Click here to see the algorithm types available for this system event.
Event Activation	Required	This algorithm is executed to activate an overdue event on its trigger date. Refer to Overdue Events Can Do Many Things and

		How and When Events Are Activated for the details. Click here to see the algorithm types available for this system event.
Event Information	Optional - only used if you want to override an overdue event's info string	This algorithm is executed to construct an overdue event's override info string. Refer to Overdue Event Information Is Overridable for the details. Click here to see the algorithm types available for this system event.
Monitor Waiting Events	Optional - only used if events of this type can enter the Waiting state	This algorithm is invoked by the Overdue / Cut Event Manager for events in the Waiting state. Refer to Some Events Wait For Something Before Completing for the details. Click here to see the algorithm types available for this system event.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference CI_OD_EVT_TYPE.

Setting Up Overdue Process Templates

An overdue process template encapsulates the business rules that govern a given type of overdue process. Open **Admin** > **Credit & Collection** > **Overdue Process Template** > **Add** to set up overdue process templates.

> **NOTE:**
> **Recommendation.** Before using this transaction, we strongly recommend that you review The Big Picture Of Overdue Processes.

Description of Page

Enter a unique **Overdue Process Template** and **Description** for the overdue process template.

Collecting On Object defines the type of object managed by this overdue process. This field actually references a foreign key characteristic type that references the managed object. For example, if this overdue process template manages overdue bills, you'd reference a foreign key characteristic that references the bill object.

The **Algorithms** grid contains algorithms that control important functions. You must define the following for each algorithm:

- Specify the algorithm's **System Event** (see the following table for a description of all possible events).
- Specify the **Algorithm** to be executed when the System Event executes. 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.

The following table describes each **System Event** (note, all system event's are optional and you can define an unlimited number of algorithms for each event).

System Event	Optional / Required	Description

Calculate Unpaid & Original Amount	Required	This algorithm is executed to calculate the unpaid and original amounts of the objects associated with the overdue process. These amounts are shown on the overdue process page and in the base-package overdue info string. Click here to see the algorithm types available for this system event.
Cancel Criteria	Required	This algorithm is executed to determine if an overdue process can be cancelled. Refer to How Are Overdue Processes Cancelled for the details. Click here to see the algorithm types available for this system event.
Cancel Logic	Required	This algorithm is executed to cancel an overdue process. Refer to How Are Overdue Processes Cancelled for the details. Click here to see the algorithm types available for this system event.
Hold Event Activation Criteria	Optional - only used if overdue processes of this type can be suspended while some condition is true	This algorithm is executed to determine if the activation of overdue and cut events should be suspended. Refer to Holding Events for the details. Click here to see the algorithm types available for this system event.
Overdue Process Information	Optional - only used if you want to override an overdue process's info string	This algorithm is executed to construct an overdue process's override info string. Refer to Overdue Process Information Is Overridable for the details. Click here to see the algorithm types available for this system event.

The **Event Types** control the number and type of overdue events linked to an overdue process when it is first created. The information in the scroll defines these events and the date on which they will be triggered. The following fields are required for each event type:

- **Event Sequence.** Sequence controls the order in which the overdue event types appear in the scroll.
- **Overdue Event Type.** Specify the type of overdue event to be created.
- **Days After.** If **Dependent on Other Events** is on, events will be triggered this many days after the completion of the dependent events (specified in the grid). Set this value to 0 (zero) if you want the event triggered immediately after the completion of the dependent events. If **Dependent on Other Events** is off, events will be triggered this many days after the creation of the overdue process. Refer to How and When Events Are Activated for the details.
- If **Dependent on Other Events** is on, define the events that must be completed or cancelled before the event will be triggered.
    - **Sequence** is system-assigned and cannot be specified or changed.
    - **Dependent on Sequence** is the sequence of the dependent event.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference CI_OD_PROC_TMP.

Setting Up Cut Event Types

A cut event type encapsulates the business rules that govern a given type of cut event. Open **Admin** > **Credit & Collection** > **Cut Event Type** > **Add** to set up cut event types.

NOTE:
Recommendation. Before using this transaction, we strongly recommend that you review Cut Events Are Like Overdue Events.

Description of Page

Enter a unique **Cut Event Type** code and **Description** for the cut event type.

Use **Long Description** to provide a more detailed explanation of the purpose of the cut event type.

The **Algorithms** grid contains algorithms that control important functions. You must define the following for each algorithm:

- Specify the algorithm's **System Event** (see the following table for a description of all possible events).
- Specify the **Algorithm** to be executed when the System Event executes. 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.

The following table describes each **System Event**.

System Event	Optional / Required	Description
Cancel Logic	Required	This algorithm is executed to cancel a cut event. Refer to How Are Events Canceled for the details. Click here to see the algorithm types available for this system event.
Event Activation	Required	This algorithm is executed to activate a cut event on its trigger date. How and When Events Are Activated for the details. Click here to see the algorithm types available for this system event.
Event Information	Optional - only used if you want to override a cut event's info string	This algorithm is executed to construct a cut event's override info string. Refer to Cut Event Information Is Overridable for the details. Click here to see the algorithm types available for this system event.
Monitor Waiting Events	Optional - only used if events of this type can enter the Waiting state	This algorithm is invoked by the Cut / Cut Event Manager for events in the Waiting state. Refer to Some Events Wait For Something Before Completing for the details. Click here to see the algorithm types available for this system event.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference CI_CUT_EVT_TYPE.

Setting Up Cut Process Templates

A cut process template encapsulates the business rules that govern a given type of cut process. Open **Admin** > **Credit & Collection** > **Cut Process Template** > **Add** to set up cut process templates.

> **NOTE:**
> **Recommendation.** Before using this transaction, we strongly recommend that you review The Big Picture Of Cut Processes.

Description of Page

Enter a unique **Cut Process Template** and **Description** for the cut process template.

The **Algorithms** grid contains algorithms that control important functions. You must define the following for each algorithm:

- Specify the algorithm's **System Event** (see the following table for a description of all possible events).
- Specify the **Algorithm** to be executed when the System Event executes. 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.

The following table describes each **System Event**.

System Event	Optional / Required	Description
Cancel Logic	Required	This algorithm is executed to cancel a cut process. Refer to How Are Cut Processes Cancelled for the details. Click here to see the algorithm types available for this system event.
Cut Process Information	Optional - only used if you want to override a cut process's info string	This algorithm is executed to construct a cut process's override info string. Refer to Cut Process Information Is Overridable for the details. Click here to see the algorithm types available for this system event.

The **Event Types** control the number and type of events linked to a cut process when it is first created. The information in the scroll defines these events and the date on which they will be triggered. The following fields are required for each event type:

- **Event Sequence.** Sequence controls the order in which the cut event types appear in the scroll.
- **Cut Event Type.** Specify the type of cut event to be created.
- **Days After.** If **Dependent on Other Events** is on, events will be triggered this many days after the completion of the dependent events (specified in the grid). Set this value to 0 (zero) if you want the event triggered immediately after the completion of the dependent events. If **Dependent on Other Events** is off, events will be triggered this many days after the creation of the cut process. Refer to How and When Events Are Activated for the details.

- If **Dependent on Other Events** is on, define the events that must be completed or cancelled before the event will be triggered.
    - **Sequence** is system-assigned and cannot be specified or changed.
    - **Dependent on Sequence** is the sequence of the dependent event.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference CI_CUT_PROC_TMP.

Setting Up Collection Class Overdue Rules

Collection class overdue rules contain algorithms that impact accounts associated with a given collection class, division and currency code are managed. Open **Admin** > **Credit & Collection** > **Collection Class Overdue Rules** > **Add** to set up collection class overdue rules.

➤ **NOTE:**
Recommendation. Before using this transaction, we strongly recommend that you review Different Overdue Rules For Different Customers.

Description of Page

Enter the **Collection Class**, **CIS Division** and **Currency Code** to which the rules apply.

The **Algorithms** grid contains algorithms that control important functions. You must define the following for each algorithm:

- Specify the algorithm's **System Event** (see the following table for a description of all possible events).
- Specify the **Algorithm** to be executed when the System Event executes. 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.
  The following table describes each **System Event**.

System Event	Optional / Required	Description
Bill-Based Payment Arrangement	Optional - only specified if your implementation uses bill-oriented payment arrangements	This algorithm is executed to handle the creation, breaking and canceling of a Bill-Oriented Payment Arrangements. Click here to see the algorithm types available for this system event.
Overdue Monitor Rule	Required	This algorithm is invoked by the Overdue Monitor to analyze an account's debt. Refer to How Does The Overdue Monitor Work for the details. If you have multiple rules (and therefore multiple algorithms), please take care when assigning the sequence number, as the Overdue Monitor will invoke these rules in sequence order. Click here to see the algorithm types available for this system event.

Write Off Bill	Option - only specified if your implementation writes-off bills	This algorithm is executed to handle the write-off and write-off reversal of a bill. Refer to Writing Off Bills. Click here to see the algorithm types available for this system event.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference CI_OD_RULE_ALG.

Setting Up Overdue Event Cancellation Reasons

An overdue event cancel reason must be supplied before an overdue event can be canceled. Open **Admin** > **Credit & Collection** > **Overdue Event Cancel Reason** to define overdue event cancellation reasons.

Description of Page

Enter an easily recognizable **Overdue Event Cancel Reason** and **Description** for each cancellation reason.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference CI_OEVT_CAN_RSN.

Setting Up Cut Event Cancellation Reasons

A cut event cancel reason must be supplied before a cut event can be canceled. Open **Admin** > **Credit & Collection** > **Cut Event Cancel Reason** to define cut event cancellation reasons.

Description of Page

Enter an easily recognizable **Cut Event Cancel Reason** and **Description** for each cancellation reason.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference CI_CEVT_CAN_RSN.

Defining Prepaid Metering Options

The topics in this section describe how to set up the system to enable prepaid metering functionality.

➤ **NOTE:**
Prepaid metering is optional. The system configuration requirements described in this section are only relevant if your organization offers prepaid metering service.

The Big Picture of Prepaid Metering

Prepaid metering allows customers to pay for energy before it is actually used.

How Does Prepaid Metering Work?

A traditional electronic prepaid metering system typically operates at three levels:

- Meters that are installed at the customer's home. These meters dispense energy up to the amount that the customer prepaid.
- Vending stations located at the utility's offices or designated payment agencies. These stations/agencies sell prepaid credits to customers.
- Master stations that group vending stations together for administration, reporting and control purposes. Master stations communicate information - e.g. customer information, tariff changes, etc - to the vending stations. The vending stations, in turn, report detailed customer sales and other transaction information to the master station.

A prepaid customer goes to a vendor and purchases credits for his/her meter to dispense energy. The vendor has knowledge of the customer and the rate that the customer is paying. The vendor issues the credits in any of the following forms: a prepaid card, a token (to be inserted into the meter) or a series of numbers (to be keyed into the meter).

Debit Meter vs. Credit Meter

Prepaid meters are commonly referred to as debit meters. Billed meter types are also called credit meters. A meter's state of being 'credit' or 'debit' translates to a specific meter configuration. Having this property set at the meter configuration level enables meters to be switched between 'credit' and 'debit' if needed.

Prepaid credits may be purchased for either new premises or existing premises. In the latter case, the existing premise could initially have a typical 'credit' meter. In this case, fieldwork may be involved in getting the meter switched from 'credit' to 'debit' (prepaid).

Prepaid Metering Transactions Result in Adjustments

There could be numerous types of prepaid transactions. But the most common types are prepaid sales and cancellations.

A sale or cancellation transaction includes a total amount that usually includes tax. In order to post to the general ledger, the prepaid transaction must result in a financial transaction. Moreover, separate GL entries must be created for the breakdown of revenue and tax. In some cases, it may also be necessary to calculate the usage from the revenue amount.

Calculated Adjustments are used to book revenue from these types of transactions. A base Generate Adjustment algorithm is available to let rate application generate the appropriate GL calculation lines and calculate the corresponding usage, if needed.

Prepaid adjustments do not affect a customer's balance.

Interfacing Prepaid Transactions

An adjustment upload batch process exists for uploading prepaid transactions that result in adjustments. Refer to Interfacing Adjustments From External Sources for more information.

Prepaid metering options need only be set up if your organization offers prepaid metering service to your customers. Refer to Defining Prepaid Metering Options for more information.

Prepaid Transactions Can Go Into Suspense

A prepaid transaction goes into suspense if a prepaid SA could not be determined for any of the following reasons:

- The badge number on the transaction is not a valid meter in the system
- The badge number on the transaction is a valid meter in the system, but is:
  - Not linked to a prepaid meter configuration type
  - Not linked to a non-closed, non-canceled prepaid SA

This can be a common occurrence because of the likelihood that transactions get uploaded prior to meter installation/ exchange or service agreement information being updated in the system.

When this situation happens, you probably still want to recognize the revenue by posting an adjustment to a suspense SA that you designate. Refer to the base sample Get Prepaid SA Using Badge Number (C1-SABYBADGE) algorithm for an example of how the suspense SA is specified.

A batch process exists for automatically resolving suspense adjustments. Refer to How Are Suspense Adjustments Resolved for more information.

Setting Up The System To Enable Prepaid Metering

The following sections describe the steps in setting up control information for prepaid metering.

Meter Types

Prepaid meters are set up just like any other meter - i.e. the meter must reference a meter type and the meter's configuration must reference a meter configuration type.

To set up prepaid meter types you must do the following:

- Define the meter configuration types that will be used for recording prepaid usage. Mark each of these configuration types as Prepaid . Define one register with the appropriate UOM (e.g. kWh).
- Define the meter types that will be used for prepaid metering. For each meter type, associate one of the Prepaid meter configuration types that you defined. You can also mark each meter type as Prepaid Capable.

➤ **NOTE:**
Base logic does not require the meter type to be marked as Prepaid Capable. The Prepaid indicator on the meter configuration type is used to identify prepaid meters.

Characteristic Types

The following characteristic types are needed if you are going to upload prepaid transactions as adjustments and need the system to determine the prepaid SA given a prepaid meter's badge number.

Refer to Setting Up Characteristic Types for more information.

Prepaid Entity Characteristic Type

Create a characteristic type that identifies whether an SA Type is used for prepaid metering. For example:

- Characteristic Type = <code>
- Description = Prepaid Entity
- Char Entities = SA Type
- Subclass = predefined list
- Values = Y, N

Badge Number Characteristic Type

Create a characteristic type for specifying a badge number. For example:

- <code>
- Description = Badge Number
- Char Entities = Adjustment, Adjustment Type
- Subclass = Adhoc

Algorithms

The following algorithms need to be set up in order to store prepaid transactions in the system.

Refer to Setting Up Algorithms for more information.

> **NOTE:**
> The following sections describe basic set up needed for uploading 'sale' and 'cancellation' types of prepaid transactions. Your implementation team may have to define additional specific algorithms types for any other prepaid transaction type that you need to upload into or store in the system.

Calculation Rule - Calculation Algorithm

Create an algorithm of type Determine Percent Given Total (C1-PCTGVNTOT) specifying your parameter values for UOM / TOU / SQI.

Create an algorithm of type Back Into Revenue (C1-BACKINREV) specifying your parameter values for UOM / TOU / SQI.

Adjustment Type - Generate Adjustment

Create an algorithm of type Adjustment Generation - Apply Rate (ADJG-RT), specifying your prepaid rate and parameter values for UOM / TOU / SQI.

Adjustment Type - Determine SA

Create an algorithm of type Get Prepaid SA Using Badge Number (C1-SABYBADGE), specifying your badge number characteristic type, prepaid characteristic type and suspense SA ID (i.e. SA to which the adjustment needs to post when a valid prepaid SA is not found).

Adjustment Type - Resolve Suspense

Create an algorithm of type Cancel Suspense Adjustment (C1-CANSUSADJ) specifying the following:

- Badge number characteristic type
- Prepaid entity characteristic type
- Prepaid adjustment type
- Adjustment cancel reason

If you want this algorithm to create to do entries for adjustments that have been in suspense for too long, specify additional parameter values for number of days and to do type and to do role (optional).

Adjustment Type - Adjustment Information

Specify the SUSPENSE_DESCR parameter on your adjustment information algorithm if you want the information to include an indication of suspense.

Installation - Adjustment Information

Specify the SUSPENSE_DESCR parameter on your adjustment information algorithm if you want the information to include an indication of suspense.

Rate

In order to generate separate calculation lines for the revenue and tax breakdown of the prepaid amount, you need to set up a prepaid rate with components that calculate revenue amount and tax amount given just a total amount.

Create a rate schedule with the following components:

- Tax Calculation Rule
  - Calculation Rule Type = Calculation Algorithm
  - Value Type = Percentage
  - Value Source = Bill Factor
  - Bill Factor = your tax bill factor
  - Calculation Algorithm = the one created for algorithm type C1-PCTGVNTOT. Refer to Calculation Rule - Calculation Algorithm for information.
  - Distribution Code = *your distribution code for tax liability*
- Revenue Calculation Rule
  - Calculation Rule Type = Calculation Algorithm
  - Derive SQ = checked
  - Value Type = Unit Charge
  - Value Source = Bill Factor
  - Bill Factor = your rate/kWh bill factor
  - UOM/TOD/SQI to use for the resulting calculated usage, for example kWh
  - RC Cross Reference = indicate the sequence for the tax calculation rule (above)
  - Calculation Algorithm = the one created for algorithm type C1-BACKINREV. Refer to Calculation Rule - Calculation Algorithm for information.
  - Distribution Code = your distribution code for revenue

Adjustment Types

You must define an adjustment type for each type of transaction that you want to store in the system. Refer to Setting Up Adjustment Types for more information.

For prepaid sales and cancellation transactions, you need to define calculated adjustment types. In addition, plug-in the following algorithms:

- Adjustment FT Creation - GL Only algorithm - so that transaction amounts do not affect the customer's balance. Specify calc lines as the source for adjustment distribution codes.
- Generate Adjustment
- Determine SA
- Resolve Suspense, if applicable.
- Adjustment Information, if applicable

Adjustment Type Profiles

Create appropriate adjustment type profiles for your prepaid service agreements.

Service Agreement Types

You must create a prepaid SA Type for each unique CIS Division. Each SA Type must be set up as non-billable and requiring a characteristic premise. The SA Types must also indicate that they are used for prepaid by specifying a prepaid entity characteristic value.

Bill Cycle

Create at least one bill cycle that will be used for prepaid accounts. This bill cycle should not have a schedule defined.

Vendor Information

A vendor could be a vending station, payment agency or master station.

You must create a person for each of your vendors. Information about the vendor may include names, an address, phone numbers, etc.

Any other miscellaneous information about the vendor can be stored as characteristics on the person.

Hierarchical relationships between vendors can be established through characteristics. For instance, a vending station's person record may contain a foreign key characteristic that points to the master station that the vending station reports to.

Defining Conservation Program Options

Oracle Utilities Customer Care and Billing allows you to define conservation (or energy efficiency) programs and provide rebates to customers.

The Big Picture of Conservation Programs

The purpose of using conservation programs is to provide rebates to customers based on eligibility and verification of newly purchased appliances and hardware that are rated to conserve the demand for energy. To redeem their rebates, customers have to submit a rebate application to the utility with receipts, and the utility has to administer and report on the programs.

Admin and transaction objects are provided in the product to support the definition of conservation programs (admin data) and the subsequent rebate claims (transaction data). Portals and BPA scripts are used to maintain the conservation programs and rebate claims. Adjustments are used to recognize the expense and to process refunds.

The following sections discuss the maintenance objects that support this functionality.

Conservation Program Maintenance Object

A conservation program is an admin maintenance object (MO) used to support the definition of a conservation program. It holds rules that control how the claims for a conservation program are managed. If your organization wishes to use this MO, you can either use the business object (BO) supplied in the base product or configure your own BO.

This MO provides the following functionality:

- A business object option Display Statistics Service Script is provided for business objects of this type. The script plugged into this option retrieves information displayed on the **Conservation Program Statistics Zone**.
- A business object option Display Statistics UI Map is provided for business objects of this type. This is the map used on the **Conservation Program Statistics Zone** to display statistics.
- A separate maintenance object is provided to capture rebate definitions for a conservation program
- Logs are not provided.
- The standard characteristics collection is provided.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference CI_CONSV_PROG.

FASTPATH:
For more information, about this MO and to review the business objects defined for this MO, navigate to **Admin** > **Database** > **Maintenance Object** > **Search** and view the MO C1-CPROG.

Conservation Program Rebate Definition Maintenance Object

A conservation program has rebate definitions that define the rebate amounts and energy savings for different types of products. The superset of such lines is the "rebate matrix". Each row in the matrix defines the rebate and presumed energy savings for a product purchased by the consumer. If your organization wishes to use this maintenance object (MO), you can either use the business object (BO) supplied in the base product or configure your own BO.

This MO provides the following functionality:

- The Rebate Definition table can be used to capture information for rebate claim lines such as, appliance type, refund amount, energy savings, and applicable manufacturers and models.
- Logs are not provided for the Rebate Definition.
- A characteristics collection is not provided.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference CI_REBATE_DEFN.

- **FASTPATH:**
  For more information, about this MO and to review the business objects defined for this MO, navigate to **Admin** > **Database** > **Maintenance Object** > **Search** and view the MO C1-RDEF.

Rebate Claim Maintenance Object

When a customer files a claim for a rebate, a rebate claim will be created. The rebate claim maintenance object (MO) is used to support the definition of a claim. If your organization wishes to use this MO, you can either use the business object (BO) supplied in the base product or configure your own BO.

This MO provides the following functionality:

- Depending on the utility's requirements, you can specify single or multiple items on the Rebate Claim. A separate maintenance object is provided to capture rebate claim lines for a rebate claim.
- The standard characteristics collection is provided.
- A log is provided. This can be used to track approval information for the claim, to capture adjustments created as the claim is processed, etc.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference CI_REBATE_CLAIM.

- **FASTPATH:**
  For more information, about this MO and to review the business objects defined for this MO, navigate to **Admin** > **Database** > **Maintenance Object** > **Search** and view the MO C1-RCLAIM.

Rebate Claim Line Maintenance Object

A rebate claim has a rebate claim line for each product eligible for a refund. You use the rebate claim line maintenance object (MO) to create a rebate line. If your organization wishes to use this MO, you can either use the business object (BO) supplied in the base product or configure your own BO.

This MO provides the following functionality:

- SA (service): The SA table is the real service SA. There is a separate SA under which the rebate adjustments are stored.
- Logs are not provided for Rebate Claim Line
- The standard characteristics collection is provided.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference CI_REBATE_LINE.

- **FASTPATH:**
  For more information, about this MO and to review the business objects defined for this MO, navigate to **Admin** > **Database** > **Maintenance Object** > **Search** and view the MO C1-REBLN.

GL Accounting Example

The following table shows the financial transactions that are issued when processing energy conservation rebates. This example shows the financial transactions when a customer files a claim for a dishwasher and insulation.

Notice how the expense adjustments are atomized:

- Separate adjustments are created to expense each item individually.
- A single adjustment is created for the A/P check request.

Event	GL Accounting
Rebate adjustment is created - dishwasher	Rebate Expense 35 A/P <35>
Rebate adjustment is created - insulation	Rebate Expense 45.45 A/P <45.45>
A/P check request issued	A/P 80.45 Cash <80.45>

Setting Up Conservation Programs

Conservation Programs allow administrators to create and maintain conservation programs for which customer can submit rebate claims. To set up a conservation program, open **Admin** > **Customer** > **Conservation Program** > **Add.**

- **FASTPATH:**

> For additional information on working with rebate claims, see Rebate Claims.

The topics in this section describe the base-package zones that appear on the Conservation Program portal.

Conservation Program List Zone

The Conservation Program List zone lists every conservation program. The following functions are available:

- Click a broadcast button to open other zones that contain more information about the adjacent conservation program.
- Click the Add link in the zone's title bar to add a new conservation program.

This zone displays the following for each conservation program defined in the system:

- Conservation Program Information (Description, Status, Start Date, End Date)
- The number of Complete Claims
- The total Complete Claims Amount
- The number of Incomplete Claims
- The Statistics Date/Time is the date that the statistics were last calculated

Use the following procedure to create a new conservation program:

- Select **Admin** > **Customer** > **Conservation Program** > **Add** or click **Add** on the title bar of the Conservation Program List zone.
- Enter basic information about the conservation program in the **Main** section, including:
  - Conservation Program code
  - Description
  - Start Date/End Date
  - Statistic Update Frequency (in hours)
- Enter **Financial Information** about the conservation program, including:
  - CIS Division
  - SA Type
  - A/P Adjustment Type
  - A/P Taxable Adjustment Type
- Enter **Approval Information** about the conservation program, including:
  - Approval To Do Type
  - Duplicate Claim To Do Role
  - Approval Levels and Threshold Amounts
- Click **Save.** To return to the Conservation Program portal without saving the new program, click **Cancel.**

To view a specific conservation program, click the broadcast icon for the conservation program you wish to view. The remaining zones in the Conservation Program portal open displaying details about the selected conservation program.

Conservation Program Zone

You use the Conservation Program zone to view and maintain individual conservation program. The Conservation Program zone displays the following information about the selected conservation program:

- Basic information about the conservation program, including
  - Conservation Program
  - Description
  - Start Date/End Date
  - Statistic Update Frequency (in hours)
  - Status
- Financial Information about the conservation program, including
  - CIS Division
  - SA Type
  - A/P Adjustment Type
  - A/P Taxable Adjustment Type
- Approval Information about the conservation program, including
  - Approval To Do Type
  - Duplicate Claim To Do Role
  - Approval Levels and Threshold Amounts

Please see the zone's help text for information about this zone's fields.

Conservation Program Status

The Status of a conservation program indicates the current state of the program within the system. Valid statuses include:

- **Pending** indicates the program is pending. This is the initial state of a conservation program when first created.
- **Active** indicates the conservation program is currently active.
- **Inactive** indicates the conservation program is currently inactive.

Conservation Program Actions

You can perform a number of actions on a conservation program, including:

- **Edit:** Used to edit a conservation program
- **Delete:** Used to delete a conservation program
- **Activate:** Used to activate a pending or inactive conservation program
- **Refresh Statistics:** Used to refresh the statistics of an active conservation program
- **Deactivate:** Used to deactivate a pending or active conservation program

- **Pend:** Used to change the status of an active or inactive conservation program to Pending

The actions available are based on the current status of the conservation program. The table below summarizes the actions available at each status.

Status	Valid Action
Pending	Edit, Delete, Activate, Deactivate
Active	Edit, Refresh Statistics, Deactivate, Pend
Inactive	Activate, Pend

Editing Conservation Programs

Use the following procedure to edit an active or pending conservation program:

- Click **Edit.**
- Edit the details of the conservation program as needed.
- Click **Save.**

Deleting Conservation Programs

Use the following procedure to delete a conservation program:

- Click **Pend** to change the status of the conservation program to Pending (if needed).
- Click **Delete.**
- Click **OK** on the Confirm Delete dialog. To close the dialog without deleting the conservation program, click **Cancel.**

Activating Conservation Programs

To activate an inactive or pending conservation program, click **Activate.**

Deactivating Conservation Programs

To deactivate an active or pending conservation program, click **Deactivate.**

Setting the Status of Conservation Programs to Pending

To set the status of an active or inactive conservation program to Pending, click **Pend.**

Rebate Definition Zone

You use the Rebate Definition zone to add, view, and edit rebate definitions associated with a conservation program. A rebate definition defines the types of rebates allowed for a specific conservation program. The Rebate Definition zone displays the following details for each claim line:

- Rebate Definition
- Status (Active or Inactive)
- Rebate Amount per Unit
- Icons to Edit, Delete, and Activate/Deactivate rebate definitions

Adding Rebate Definitions

Use the following procedure to add a rebate definition:

- Click **Add** in the title bar of the Rebate Definition zone.
- Enter a Description of the rebate definition.
- Select the **Product** for the rebate definition from the dropdown list.
- Select the **Service Type** for the rebate definition from the dropdown list.
- Enter the **Rebate Per Unit Amount** for the rebate definition.
- Enter the **Rebate Unit of Measure** for the rebate definition.
- Select the **Expense Adjustment Type** for the rebate definition from the dropdown list.
- Enter the **Presumed Energy Savings** for the rebate definition.
- Select the **Presumed Energy Savings** Unit of Measure for the rebate definition from the dropdown list.
- Enter the **Manufacturer** and **Model** for each item eligible to be submitted as a rebate claim line.
- To upload manufacturer/model information from a comma-separated-values file, click **CSV File to Upload.**
    - Click **Browse** on the File Upload dialog, and browse to the file to be uploaded.
    - Click **Upload.**
- Click **Save.** To return to the Rebate Definition zone without adding the rebate definition, click **Cancel.**

Editing Rebate Definitions

Use the following procedure to edit a rebate definition:

- Click the edit icon for the rebate definition you wish to delete.
- Edit the details of the claim line as appropriate.
- Click **Save**. To return to the Rebate Definition zone without changing the rebate definition, click **Cancel.**

> **NOTE:**
> You can edit a rebate definition only when it is Active.

Deleting Rebate Definitions

Use the following procedure to delete a rebate definition:

- Click the delete icon for the rebate definition you wish to delete.
- Click **OK** on the Confirm Delete dialog. To close the dialog without deleting the rebate definition, click **Cancel.**

Activating Rebate Definitions

To activate an inactive rebate definition, click the **Activate** button for the rebate definition you wish to activate.

> ➤ **NOTE:**
> You can only activate rebate definition that is currently Inactive.

Deactivating Rebate Definitions

To deactivate an active rebate definition, click the **Deactivate** button for the rebate definition you wish to deactivate.

> ➤ **NOTE:**
> You can only deactivate rebate definition that is currently Active.

Rebate Claim Statistics Zone

You use the Rebate Claim Statistic zone to view statistics for submitted rebate claims based on the current conservation program. This zone displays the following:

- Conservation Program
- Statistics Date/Time
- Statistics Charts

Statistics Graphs

This zone displays statistics for submitted rebate claims using the following charts:

- **Statistics by Status:** A pie chart that displays the number and percentage of claims for each status.
- **Statistics by Month:** A line chart that displays the number of claims completed each month.
- **Statistics by Product:** A pie chart that displays the total value, percentage, and number of claim lines completed for each product.

- **Statistics by Claim Age:** A line and bar chart that displays the number of new claims per month (line graph) and the number of claims per age grouping (bar graph).

To refresh statistics, click **Refresh Statistics** on the title bar of the Rebate Claim Statistics zone.

Defining Market Options

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.

Configuring Markets

This portal is used to display and maintain a Market.

Refer to Understanding Markets for more information.

You can access the portal from the **Admin** > **Open Market** > **Market** > **Search.**

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.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference D1_MKT.

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 service provider 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 from the **Admin** > **Open Market** > **Market Participant** > **Search.**

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.
- **Service Provider:** This zone provides information about the selected Market Participant.
- **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.
- **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.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference D1_SPR.

Configuration

This section contains additional configuration topics for Customer Care and Billing.

Configuring Zones

Many zones in Oracle Utilities Customer Care and Billing do not require configuration by your implementation team. For example, the base package is shipped with the Account Financial History zone that appears on the Control Central - Account Information portal. This zone does not require configuration because its zone type has no configurable options (i.e., its behavior is static).

Other zones require configuration before they can be used because their behavior is dynamic. The topics in this section provide tips and techniques on how to configure zones in Oracle Utilities Customer Care and Billing.

FASTPATH:
Refer to The Big Picture of Portals and Zones in the *Oracle Utilities Application Framework Administration Guide* for a description of portal and zone functionality.

Configuring Timeline Zones

A timeline zone is a zone that may be configured on the account or customer information tabs of Control Central. This type of zone contains one or more "lines" where each line shows when significant events have occurred. For example, you can set up a timeline zone that has two lines: one that shows when payments have been received from a customer, and another that shows when bills have been sent to the customer.

FASTPATH: For a complete description of the numerous features available on a timeline zone, refer to Timeline Zone.

Configuration is required to enable a timeline zone. A timeline zone is made up of one or more algorithms where each algorithm returns “events” for a certain type of record (payment or bill for example). The product delivers several algorithm types for the timeline zone to include “events” for many different transactions in the product. Each algorithm type uses the Account or Person in context to retrieve the related information.

- Refer to Configuring Timeline Zones in the framework portion of the administration guide for information about the type of detail each event may return. The base provided algorithm types include many parameters to allow the implementation to configure much of this information as desired.
- Click here to see the algorithm types available for this plug-in spot.

Many of the algorithms support supplying a BPA script to launch when a user clicks on an event on a timeline. Be sure to review each algorithm type to see what types of events support a BPA script. For each event, determine if your business practice warrants the development of a BPA script.

After configuring the algorithms per your business needs, the following additional steps are needed:

- Set up a zone that references these algorithms. The zone will reference the **F1-TIMELINE** zone type.
- Link the zone to the appropriate portal(s) (e.g., Control Central - Account Information or Control Central - Customer Information).
- Update your users' portal preferences and security rights so they can see the zone in the desired location on the portal(s).

You can set up many timeline zones. For example,

- You might want different zones to appear on a portal depending on the type of user. For example, you might want one timeline for billing clerks, and a different one for customer service representatives.
- For aesthetic reasons, you might want multiple simple timeline zones to appear on a given portal rather than one complex timeline zone.
- You might want to set up context specific timeline zones. For example, you might want to have one timeline zone that is premise-oriented and another that is person-oriented.

Configuration Migration Assistant (CMA) Addendum

This section is an addendum to the general Configuration Migration Assistant section in the *Oracle Utilities Application Framework Administration Guide*. This section assumes that you are familiar with the concepts of Configuration Migration Assistant (CMA).

> **NOTE:** CMA is designed to migrate configuration data only. Since base product data should be updated only through product patching mechanisms, CMA cannot currently be used to migrate master/transactional data that contains system-generated primary keys, such as customer account or billing information.

This addendum describes Configuration Migration Assistant (CMA) functionality that is specific to Oracle Utilities Customer Care and Billing.

The following sections provide information about what is provided in the C1-owned base package:

Base Package Migration Plans

The C1-owned base package provides a large number of migration plans to support migrating configuration and/or administration data from one environment to another.

Use the following procedure to access the base package migration plans:

1. Navigate to **Admin** > **Implementation Tools** > **Migration Plan** > **Search**.
2. Enter “C1” in the **Migration Plan** field.
3. **Click Refresh**.
4. Select a migration plan in the search results list. The details of the selected migration plan are displayed in the Migration Plan portal.

> **NOTE:** The C1-owned base package plans can also be used as a basis for custom migration plans. To create a custom migration plan, select a plan to base the custom migration plan on, click the Duplicate button, and

define the custom plan to meet the implementation's requirements as described in Defining a Migration Plan in the *Oracle Utilities Application Framework Administration Guide*.

Base Package Migration Request

The C1-owned base package provides two migration request to support migrating configuration and/or administration data from one environment to another.

Use the following procedure to access the base package migration requests:

1. Navigate to **Admin** > **Implementation Tools** > **Migration Request** > **Search**
2. Enter "C1" in the **Migration Request** field.
3. Click **Refresh.**
4. Click the **Copy and Sync Control Data** link in the search results list.

The details of the migration requests are outlined below. Please use the Migration Request portal to view additional details about this migration request.

Base Admin Data Migration Request

- **Migration Request:** C1-AdminBasic
- **Description:** Base Admin Data
- **Detailed Description:** Base Admin Data.
- **Migration Plans:** This migration request includes base package migration plans used by Oracle Utilities Customer Care and Billing.

Copy and Sync Control Data Migration Request

- **Migration Request:** C1-CopyControlTables
- **Description:** Copy and Sync Control Data
- **Detailed Description:** Copy and Sync Control Data.
- **Migration Plans:** This migration request includes base package migration plans used by Oracle Utilities Customer Care and Billing.

For additional information about these migration plans, see Base Package Migration Plans.

➤ **NOTE:** The C1-owned base package request can also be used as a basis for custom migration requests. To create a custom migration request, select the base package request, click the Duplicate button, and define the request as to meet the implementation's requirements as described in Defining a Migration Request in the *Oracle Utilities Application Framework Administration Guide*.

Wholesale and Piecemeal Migrations

There are two general types of migrations used with the Configuration Migration Assistant: wholesale migrations and piecemeal migrations.

Wholesale Migrations

Wholesale migrations are used when migrating all the configuration and/or administration data from one environment to another. For example, a wholesale migration might be used when migrating admin data from a development or test environment to a production environment. For more on this type of migration, see Wholesale Migrations in the *Oracle Utilities Application Framework Administration Guide*.

The following is a high-level overview of the steps involved when executing a wholesale migration.

1. Process the “F1-SchemaAdmin” (FW Foundation) migration request (This request contains migration plans for Field, Lookup, Char Type, Currency Code and FK Ref).
2. Process the “C1-AdminBasic” migration request. This includes copies of framework migration plans (including plans for Business Objects, Algorithms, and Feature Configurations) from the “F1-FrameworkAdmin” migration request, as well as independent base package C1-owned wholesale migration plans which can be run first.
3. Process any of the other delivered framework-based (F1-owned) migration requests as needed (except for the “F1-FrameworkAdmin” migration request which is already incorporated in #2)
4. Process the “C1-CopyControlTables” migration request. This includes base package C1-owned wholesale migration plans with dependency on “C1-AdminBasic” migration request.

Piecemeal Migrations

Piecemeal (or "non-wholesale") migrations are used when migrating a small portion (or piece) of configuration and/or administration data from one environment to another. For example, a piecemeal migration might be used when migrating groups and rules from a development or test environment to a production environment. For more on this type of migration, see Piecemeal Migrations in the *Oracle Utilities Application Framework Administration Guide*.

The C1-owned base package does not contain piecemeal migration plans for Oracle Utilities Customer Care and Billing.

Processing Notes for Specific Objects

The following limitations apply to certain Oracle Utilities Customer Care and Billing objects when using Configuration Migration Assistant:

- An Issuing Center object references a User. If this user does not exist in the target system, Configuration Migration Assistant cannot apply the requested changes.
- A Case Type object references an Application Service. If this service does not exist in the target system, Configuration Migration Assistant cannot apply the requested changes.
- Collection Agency and Service Provider objects reference a Person. If this person does not exist in the target system, Configuration Migration Assistant cannot apply the requested changes.
- Service Provider and Tender Source objects reference a Service Agreement. If this service agreement does not exist in the target system, Configuration Migration Assistant cannot apply the requested changes.
- If your migrateable object includes log tables, you may add your log entity to the characteristic type F1-MGO. You should also mark the log table as a “Non-Migrateable Table” in the Maintenance Object options.
- Transactions are applied in an unspecified order (and probably in order numerically by a randomly generated ID value). CMA only looks at "hard" constraints when determining what to put into the same transaction. Any "soft" constraints such as characteristics and algorithm parameters that might have FK references to other objects are not processed by CMA. Unless the migration plan ensures that related items go into the same transaction, they will end up in different ones, and must need to apply again and again until eventually all gets applied.

To Do Lists Addendum

This section is an addendum to the general To Do Lists chapter. This addendum describes the To Do functionality that is specific to Oracle Utilities Customer Care and Billing.

Assigning A To Do Role

As described in To Do Entries Reference A Role, each To Do entry requires a role. To Do entries created in Oracle Utilities Customer Care and Billing may attempt to assign a role based on an account management group or division if it is applicable to the type of data related to the To Do entry.

As described in The Big Picture of To Do Lists, users are informed that something requires their attention by entries that appear in a To Do List. For example, consider what happens when billing can't find a reading (and it's not allowed to estimate):

- The billing process creates a bill segment that is in error (meter read cannot be found).
- This bill segment that's in error, in turn, triggers the creation of a To Do entry.
- The To Do entry is assigned a role. A role is one or more users who can look at / work on the entry.
- When users view a To Do List, they only see entries addressed to roles to which they belong.

You can optionally use account management groups (AMG) to define the respective role to be assigned to To Do entries that are associated with an account and To Do type. For example, you can create an AMG called Credit Risks and assign this to accounts with suspect credit. Then, whenever an account-oriented To Do entry is created for such an account, it will be assigned a role based on the Credit Risks AMG. Refer to Setting Up Account Management Groups for more information.

By assigning an AMG to an account, you are telling the system to address this account's To Do list entries to the roles defined on the AMG (note, each To Do type can have a different role defined for it on an AMG).

You can optionally use division to define the respective role to be assigned to To Do entries that are associated with an account and To Do type. For example, you may have a division called California Operations and assign this to accounts located in California. Then, whenever an account-oriented To Do entry is created for such an account, it will be assigned a role based on the California Operations division. Refer to CIS Division Portal for more information.

A To Do Pre-Creation installation options plug-in is provided to determine the appropriate To Do Role for an account based on AMG and division setup. If plugged in, the logic to determine To Do role for an account is performed whenever a To Do entry is created. Refer to C1-TDCR-DFRL for further details on how this plug-in works.

FASTPATH:
Refer to To Do Entries Reference A Role for the details of how an initial role is assigned to To Do entries.

System To Do Types

NOTE:

List of available To Do types. The To Do types available with the product may be viewed in the application viewer 's To Do type viewer. In addition if your implementation adds To Do types, you may regenerate the application viewer to see your additions reflected there.

Background Processes Addendum

This chapter is an addendum to the general Defining Background Processes chapter. This addendum describes the background processes that are provided with Oracle Utilities Customer to Meter.

NOTE: List of system background processes. The list of background processes provided in the base product may be viewed in the application viewer's batch control viewer. In addition if your implementation adds batch control records, you may regenerate the application viewer to see your additions reflected there.

How To Set Up A New Extract Processes

Several background processes delivered with the system are used to interface information out of the system. The topics in this section describe when and how to introduce an additional extract process.

Setting Up Meter Read Extracts

You will need a meter read extract for every mechanism your company uses to route meter read requests to the software that handles your meter reading requests. For example:

- You will need a meter read extract to interface records to your handheld device software. The MDL process delivered with the system is intended to be used to handle this function. This process will have to be populated to format the output records in keeping with the needs of your meter reading software.
- If you interface some meter read requests to automatic meter reading software, you will need a new meter read extract process.

If you need additional meter read extract processes, set up the following information:

- Add a new batch control record. Populate the fields as follows:
- **ID.** Assign an easily recognizable unique ID for the meter read extract process.
- **Description.** Enter a description of the meter read extract process.
- **Accumulate All Instances.** Turn this switch on.
- Use Route Type to define the meter read extract process to be used for each route type.
- It may be necessary to register the process with your scheduler software.

Setting Up Automatic Payment Extracts

You will need an automatic payment extract for every mechanism your company uses to route automatic payment requests to a financial institution / clearing house. For example:

- You will need an automatic payment extract to interface records to the Automated Clearing House (ACH) if you allow customer to pay via credit card or direct debit from a checking account. The **APAYACH** and C1-APACH processes delivered with the system are intended to be used to handle this function.

If you need additional automatic payment extract processes, set up the following information:

- Add a new batch control record. Populate the fields as follows:
- **Batch Process.** Assign an easily recognizable unique ID for the automatic payment extract process.
- **Description.** Enter a description of the automatic payment extract process.
- **Accumulate All Instances.** Turn this switch on.
- Use Auto Pay Route Type to define the auto pay extract process to be used for each route type.

Choosing Which Records Are Selected

The system is delivered with some background processes that utilize the **Select Records** plug-in spot to select the records to be processed. It is important to note that these background processes do not utilize the Process Records plug-in spot. This logic is hardcoded as these background processes are some of the core system functions such as billing and collections. This functionality provides flexibility in choosing which records are processed, but not how they are processed.

The background processes delivered with the system optionally selects records based on CIS Division. One example is the batch control (CI-BILL) and algorithm (C1-BILL-SR). The background process and select records algorithm work as a pair. The batch control defines the parameter to be used by the select records algorithm. The algorithm defines the thread strategy, key fields, and makes the batch control parameter available to be used in the SQL algorithm parameter. Click here to see the algorithm types available for the Select Records system event.

Customizing Which Records Are Selected

There are two ways to customize what is provided with the system.

If your organization processes records using the same criteria as the samples delivered in the system, in this case CIS Division, simply create a custom batch control for each CIS Division and specify the corresponding parameter value.

If your organization processes records using different criteria, the following points summarize the steps needed to implement a new background process using plug-ins for the specific functionality:

- Duplicate an existing batch control
- Remove the batch parameter not relevant to your organization and replace with the relevant batch parameter. Note: only the batch parameters associated with selecting records should be removed or updated. Parameters uses to process records must remain.
- Duplicate the plug-in script associated with the select records algorithm type. Modify the script to pass the new batch parameter(s) to bind variables and remove those that are no longer batch parameters.
- Duplicate the select records algorithm type and associate with the new script.
- Create an algorithm for the new algorithm type. The SQL parameter value should be copied from the original algorithm type’s algorithm parameter value and modified to make use of the newly added bind variables. (**Note:** Do not change the fields selected or their order.)
- Update the batch control to use the new select records algorithm.

The Big Picture of Sample & Submit

Sample and Submit refers to the ability to create Activity Requests. This is functionality that enables an implementer to design an ad-hoc batch process using the configuration tools.

Some examples of such processes are:

- Send a letter to customers that use credit cards for auto pay and the credit card expiration date is within 30 days of the current date.
- Stop auto pay for customers that use credit cards as the form of payment if the credit card has already expired. Notify the customer that their auto pay agreement has been terminated and that they need to call to reinstate.
- Select auto pay accounts that have more than X non-sufficient fund penalties, stop the auto pay agreement and notify the customer.

NOTE:
The terms *activity request* and *sample & submit request* may be used interchangeably.

Activity Request Type Defines Parameters

For each type of process that your implementation wants to implement, you must configure an activity request type to capture the appropriate parameters needed by the activity request.

Preview A Sample Prior To Submitting

To submit a new activity request, a user must select the appropriate activity request type and enter the desired parameter values, as applicable.

After entering the parameters, the following actions are possible

- 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 **Save** to submit the request, go **Back** to adjust the parameters or **Cancel** the request.
- Click **Cancel** to cancel the request.
- Click **Save** to skip the preview step and submit the request.

When an activity request is saved, the job is not immediately submitted for real time processing. The record is save in the status Pending and a monitor process for this record's business object is responsible to transition the record to Complete.

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 activity request, such as generating customer contact records to send letters to a set of customers, is performed when transitioning to Complete (using an enter processing algorithm for the business object).

Credit Card Expiration Notice

The base product supplies a sample process to find customers that use credit cards for auto pay and their credit card expiration date is within *X* days of the current date.

To this functionality the following configuration tasks are needed:

- Define an appropriate customer contact class and type to use.
- Define appropriate activity request Cancellation Reasons. Cancellation reasons are defined using a customizable lookup. The lookup field name is C1_AM_CANCEL_RSN_FLG.
- Define an activity request type for the business object C1-NotifyExpiringCreditCardTyp . You may define default parameter values for the number of days for expiration and customer contact class and type.

Exploring Activity Request Data Relationships

Use the following links to open the application viewer where you can explore the physical tables and data relationships behind the activity request functionality:

- Click C1-ACM-ACTTY to view the activity request type maintenance object's tables.
- Click C1-ACM-ACTRQ to view the activity request maintenance object's tables.

Defining a New Activity Request

To design a new ad-hoc batch job that users can submit via Sample and Submit, first create a new Activity Request Type business object. The base product BO for the activity type C1-NotifyExpiringCreditCardTyp may be used as a sample.

The business object for the activity request includes the functionality for selecting the records to process, displaying a preview map for the user to review and to perform the actual processing. The base product BO for the activity request C1-NotifyExpiringCreditCardReq may be used as a sample. The following points highlight the important configuration for this business object:

- Special BO options are available for the activity request BOs to support the Preview Sample functionality.
  - Activity Request Preview Service Script. This script is responsible for retrieving the information displayed when a user asks for a preview of a sample of records.
  - Activity Request Preview Map. This is the map that is invoked to display the preview sample results.
- The enter algorithm plugged into the Complete state is responsible for selecting all the records that satisfy the criteria and processing the records accordingly.

Setting Up Activity Request Types

Activity Request Types define the parameters to capture when submitting an activity request through Sample and Submit. To set up an activity request type, open **Admin** > **Customer** > **Activity Request Type** > **Add.**

The topics in this section describe the base-package zones that appear on the Activity Request Type portal.

Activity Request Type List

The Activity Request Type List zone lists every activity type. The following functions are available:

- Click a broadcast button to open other zones that contain more information about the adjacent activity type.

- Click the Add link in the zone's title bar to add a new activity type.

Activity Request Type

The Activity Request Type zone contains display-only information about an activity request type. This zone appears when an activity request type has been broadcast from the Activity Request Type List zone or if this portal is opened via a drill down from another page. The following functions are available:

- Click the **Edit** button to start a business process that updates the activity request type.
- Click the **Delete** button to start a business process that deletes the activity request type.
- Click the **Duplicate** button to start a business process that duplicates the activity request type.
- State transition buttons are available to transition the activity request type to an appropriate next state.

Please see the zone's help text for information about this zone's fields.

Maintaining Sample & Submit Requests

Use the Sample and Submit transaction to view and maintain pending or historic activity requests. Navigate using **Menu** > **Tools** > **Sample & Submit Request** > **Search.**

Sample & Submit Request Query

Use the query portal to search for an existing sample & submit request. Once a request is selected, you are brought to the maintenance portal to view and maintain the selected record.

Sample & Submit Request Portal

This portal appears when a sample & submit request has been selected from the Sample & Submit Request Query portal.

The topics in this section describe the base-package zones that appear on this portal.

Sample & Submit

The Sample & Submit zone contains display-only information about an activity (sample & submit) request. The following functions are available:

- Click the **Edit** button to modify the parameters. Refer to Preview A Sample Prior to Submitting for more information.
- If the activity request is in a state that has valid next states, buttons to transition to each appropriate next state are displayed.

Please see the zone's help text for information about this zone's fields.

Sample & Submit Log

This is a standard log zone.

Security Addendum

This chapter is an addendum to the general Defining Security and User Options chapter. This addendum describes security functionality that is specific to Oracle Utilities Customer Care and Billing.

Implementing Account Security

CAUTION:
This section assumes you understand The Big Picture of Row Security.

When you create an account, you must define which users can access the account's information. For example,

- If you have customers in two geographic territories, you may need to restrict access to accounts based on the office that manages each territory. For example, only users in the northern office may manage accounts in the northern territory.
- If you have industrial and residential customers, you may need to restrict access to these different customer segments based on the skill set of the users. For example, some users are skilled in dealing with industrial customers, while others are skilled in dealing with residential customers.

By granting a user access rights to an account, you are actually granting the user access rights to the account's bills, payment, adjustments, orders, etc.

FASTPATH:
Refer to If You Do Not Practice Account Security for setup instructions if your organization doesn't practice account security.

NOTE:
Account security may also affect persons and premises. Refer to Persons Can Also Be Secured for how access to person information is also restricted by account security. Refer to Premises Can Also Be Secured for how access to premise information is also restricted by account security.

The topics in this section describe how to implement account security.

Persons Can Also Be Secured

It's important to be aware that persons can also be secured as a result of "account security". It works like this:

- If a person is linked to at least one account, users will not be allowed access to the person (or the person's related information) unless they have access to at least one of the person's accounts.

- If a person is not linked to any accounts (a rare situation), any user may access the person.

> **NOTE:**
> **How are persons linked to accounts?** A person is linked to an account when an account is created using the methods described under How To Add A New Customer From Control Central and Order User Interface Flow. In addition, you may manually link and unlink persons from account using the Account - Person page.

Premises Can Also Be Secured

It's important to be aware that premises can also be secured as a result of "account security". It works like this:

- If a premise is linked to at least one account, users will not be allowed access to the premise (or the premise's related information) unless they have access to at least one of the premise's accounts.
- If a premise is not linked to an account (a rare situation), then all users may access the premise.

> **NOTE:**
> **How are premises linked to accounts?** A premise is indirectly linked to an account. For the purpose of access restriction, we deem a premise as being linked to an account if at least one of its service points is linked to at least one of the account's service agreements.

Data Becomes Invisible When Access Is Restricted

The following summarizes the impact of a user not having access to an account:

- Account Security and Control Central
- Account Security and Searches (and Maintenance Pages)
- Account Security and To Do Lists

Account Security and Control Central

This section summarizes the impact of account security on Control Central:

- Searches are affected as follows:
- An account will only be visible if a user has access to the account's access group.
- Persons that are not linked to accounts will be visible to all users.
- If a person is linked to an account, the person will only be visible if the user has access to at least one of the person's accounts.
- Premises that are not linked to accounts will be visible to all users.
- If a premise is linked to an account, the premise will only be visible if the user has access to at least one of the premise's accounts.
- The alerts that highlight the existence of "multiple relationships" are not impacted by account security. Specifically:

- The alert Person has multiple accounts will appear if the selected person is linked to multiple accounts, even if the user doesn't have access to every account. Note well, the person couldn't have been selected if the user didn't have access rights to at least one account.
- The alert Premise has multiple accounts will appear if the selected premise is linked to multiple account, even if the user doesn't have access to every account. Note well, the premise couldn't have been selected if the user didn't have access rights to at least one account.
- Only accounts to which the user has access will be displayed in the person tree.
- Only accounts to which the user has access will be displayed in the account tree.
- All other pages contain information related to Control Central's current account context. The current account context can never reference an inaccessible account and therefore these pages are not impacted by account security.

> **NOTE:** If your implementation wishes to allow users to search for all accounts through Control Central without validating the user’s access rights, this can be configured by setting up the **Search All Account** option type on the Customer Information Options Feature Configuration . If the user tries to select an account without having the required access, they will not be able to navigate to the Account Information tab on Control Central for the selected account.

Account Security and Searches (and Maintenance Pages)

Searches are the gateway to the information that appears on maintenance pages. In general, account-related information is suppressed when a user doesn't have access rights to the account. This suppression is true for rows that directly reference an account AND for rows that indirectly reference an account. For example:

- A user can only see bills associated with accounts to which they have access rights.
- A user can only see financial transactions associated with service agreements that are, in turn, associated with accounts to which they have access rights.

> **NOTE:**
> **Person and premise searches are also impacted.** Keep in mind that information will be suppress from both person and premise-oriented searches if the person / premise is related to accounts. Refer to Persons Can Also Be Secured for how access to person information is also restricted by account security. Refer to Premises Can Also Be Secured for how access to premise information is also restricted by account security.

Account Security and To Do Lists

Account security does NOT impact the information that appears in a user's To Do list. Rather, we have assumed that your To Do roles (and the users assigned to these roles) are consistent with your account security requirements. This can result in anomalies. For example, it's possible for a supervisor to assign a bill segment error to a user who doesn't have access to the bill segment's account. This user will then see the related To Do entry in their Bill Segments In Error To Do list. However, when they drill down on the entry, account security will manifest itself (i.e., the user won't be able to display the bill segment that's in error). This happens because the drill down causes the bill segment search logic to execute. This logic inhibits the selection of bill segments if the user can't access the related account.

To minimize these anomalies, we recommend the following:

- Setup To Do Roles consistent with your Data Access Roles.

- Setup Account Management Groups that are consistent with your Access Groups.
- Setup default To Do Roles on your Account Management Groups for each To Do type.

Restricted Transactions

The following table lists all transactions that have some type of account security. The following notation is used to describe the type of account security:

- **Account-oriented.** This notation is used if the respective transaction uses basic account security (i.e., the user must belong to at least one data access role that has access to the account's access group in order to see the information).
- **Person-oriented.** This notation is used if the respective transaction uses person-oriented account security. Refer to Persons Can Also Be Secured for more information.
- **Premise-oriented.** This notation is used if the respective transaction uses premise-oriented account security. Refer to Premises Can Also Be Secured for more information.
- None of the above. Some unusual transactions have unusual implementations of account security. These are described below.

Transaction	Type of Account Security
Account	Account-oriented
Account Bill / Payment History	Account-oriented
Account Financial History	Account-oriented
Account Interval Info	Account-oriented
Account Payment History	Account-oriented
Account Person Replicator	Account-oriented
Account SAs for Debt Class	Account-oriented
Adjustment	Account-oriented
Adjustment Calculation Line Characteristics	Account-oriented
Appointment	Premise-oriented
Bill	Account-oriented
Bill Print Group	Person-oriented
Bill Segment	Account-oriented
Billable Charge	Account-oriented
Budget Review	Account-oriented
Case	Account-oriented, Person-oriented and Premise-oriented
Collection Agency Referral	Account-oriented
Collection Process	Account-oriented
Contract Option	Account-oriented
Control Central	Account-oriented, Person-oriented and Premise-oriented
Customer Contact	Person-oriented
Cut Process	Account-oriented
Declaration	Account-oriented
Deposit Review	Account-oriented

Field Activity	Premise-oriented
Field Order	Premise-oriented
Financial Transaction	Account-oriented
Financial Transaction on a Bill	Account-oriented
Financial Transaction on a Payment	Account-oriented
Interval Profile	Account-oriented (for SA-specific profiles)
Landlord Agreement	Account-oriented
Loan	Account-oriented
Match Event	Account-oriented
Multi-Cancel/Rebill	Account-oriented
Non-Billed Budget	Account-oriented
Order	Account-oriented, Person-oriented and Premise-oriented
Overdue Process	Account-oriented
Pay Plan	Account-oriented
Payment	Account-oriented
Payment Arrangement	Account-oriented
Payment Arrangement for Bills	Account-oriented
Payment Event	The user must have access to ALL accounts linked to the payment event.
Payment Event Quick Add	The user must have access to ALL accounts linked to the payment event(s).
Payment Quick Add	Account-oriented
Payment / Tender Search	Account-oriented
Person	Person-oriented
Premise	Premise-oriented
Premise Management	Premise-oriented
Quote	Account-oriented
SA Billing History	Account-oriented
SA Cash Accounting Balance	Account-oriented
SA Financial History	Account-oriented
SA Relationship	Account-oriented
Service Agreement	Account-oriented
Service Credit Event	The user must have access to ALL accounts linked to the service credit membership that has the service credit event.
Service Credit Membership	The user must have access to ALL accounts linked to the service credit membership.
Service Provider SA Relationship	Account-oriented
Severance Process	Account-oriented
Start/Stop	Account-oriented
Statement	The user can see the statement if they have access to at least one account on the statement's statement construct.

Statement Construct	The user can see the statement if they have access to at least one account on the statement construct.
Terms of Service	Account-oriented (User must have access to at least one account linked to the SA collection in the TOS.)
TOU Map	Account-oriented (for SA-specific TOU maps)
Umbrella Agreement	Account-oriented (User must have access to at least one account linked to the SA collection in the UA's TOS.)
Write Off	Account-oriented
Write Off Process	Account-oriented

Account Security Case Studies

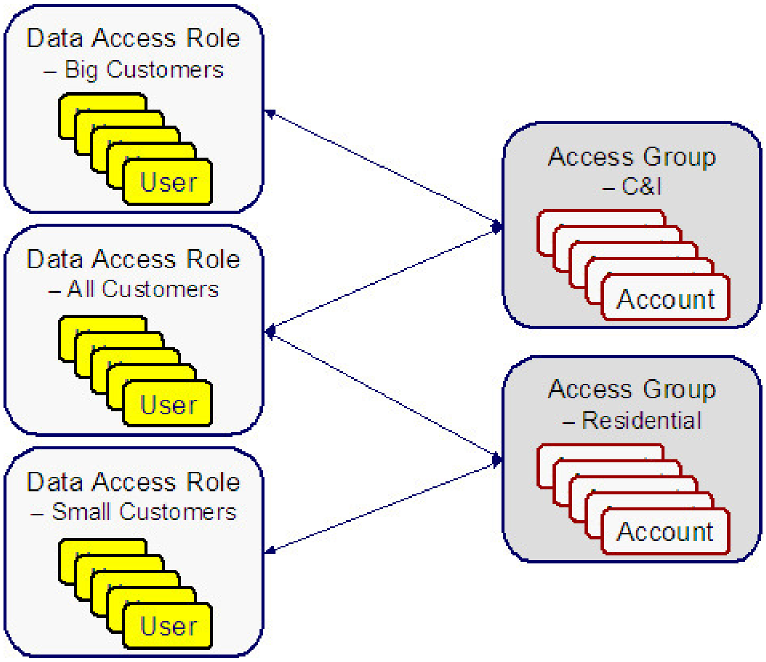
The topics in this section contain examples of how to implement account security. Use these examples to form an intuitive understanding of these objects. Once this intuition is obtained, you'll be ready to design the account security objects for your own company.

Securing Accounts Based On Customer Class

Assume the following security requirement exists:

- You have two broad groups of accounts:
- Residential accounts.
- Commercial / Industrial accounts.
- Users can be classified as have one of the following access rights:
    - May access all accounts.
    - May only access residential accounts.
    - May only access commercial / industrial accounts.

The following diagram illustrates the access groups and data access roles required to implement these requirements:



Notice the following about the above:

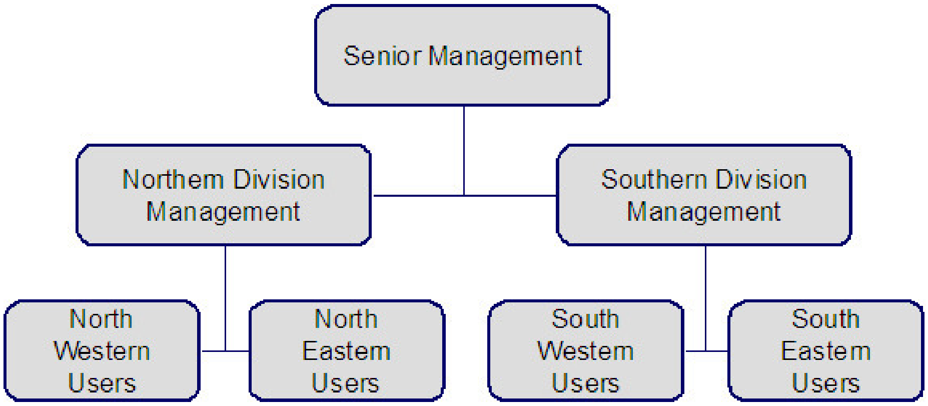
- There are 2 access groups because access to accounts is based on whether the account is considered to be residential or commercial/industrial.
- The Big Customers data access role is only linked to the C&I access group.
- The Small Customers data access role is only linked to the Residential access group.
- The All Customers access role is linked to both the C&I and Residential access groups. Users with this role can therefore access all accounts.

Securing Accounts Based On Region

Assume that accounts are classified as belonging to one of the following regions:

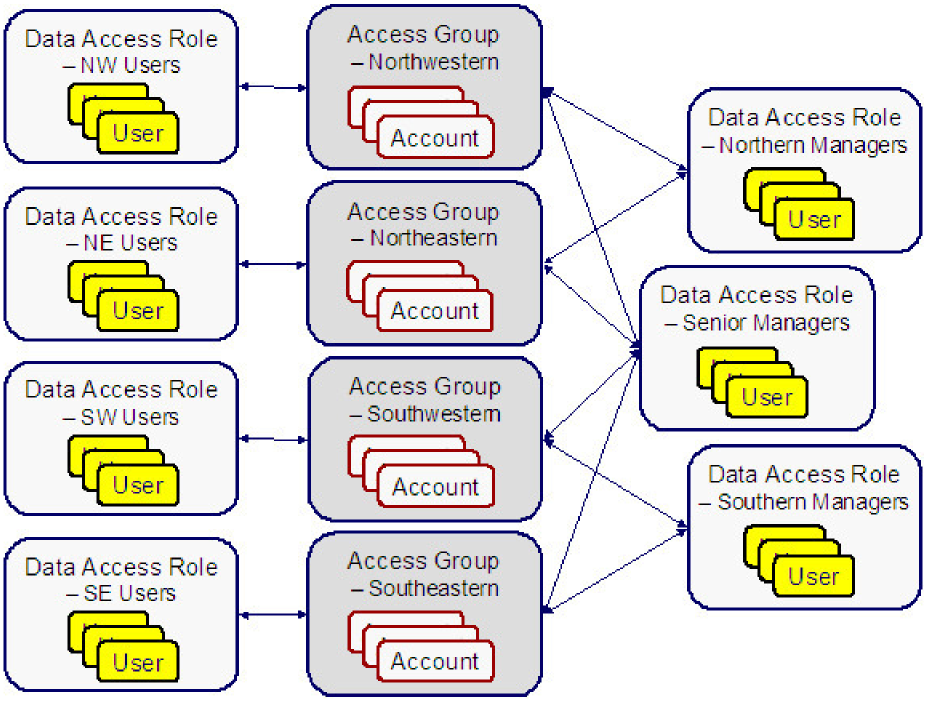
- Northwestern
- Northeastern
- Southwestern
- Southeastern

Assume the following company hierarchy exists:



- Senior Management has access to all customers
- Northern Division Management has access to all customers in the Northwestern and Northeastern divisions.
- Southern Division Management has access to all customers in the Southwestern and Southeastern divisions.
- Northwestern Users have access to all customers in the Northwestern division.
- Northeastern Users have access to all customers in the Northeastern division.
- Southwestern Users have access to all customers in the Southwestern division.
- Southeastern Users have access to all customers in the Southeastern division.

The following diagram illustrates the access groups and data access roles required to implement these requirements:


Data Access Role – NW Users
User
Access Group – Northwestern
Account
Data Access Role – Northern Managers
User
Data Access Role – NE Users
User
Access Group – Northeastern
Account
Data Access Role – Senior Managers
User
Data Access Role – SW Users
User
Access Group – Southwestern
Account
Data Access Role – Southern Managers
User
Data Access Role – SE Users
User
Access Group – Southeastern
Account

Notice the following about the above:

- There are 4 access groups because access to accounts is based on the region in which they are located (and there are 4 regions).
- There are 7 data access roles because each component of every layer of the access hierarchy requires a separate data access role.

The Default Access Group

There are two ways an access group can be assigned to an account:

- The base package will default an account's access group based on the user who adds the account. It uses the user's default access group to do this.
- If you can conceive of a rule to assign an appropriate access group to a newly added account, you can have your programming staff introduce a user exit to the account row program to implement this rule. For example, a user exit could be developed that assigns an access group to an account based on its customer class. The name of the user-exit is AFTER-MOVE-CORR-ADD and the name of the account row program is CIPCACCR.

WARNING:
Regardless of the method used to assign an account's access group, please be aware that the user who adds an account must have access to this access group.

➤

NOTE:
Subsequent changes to an account's access group. A user may change an account's access group to any access group to which they have access.

If You Do Not Practice Account Security

If you do not restrict access to accounts (i.e., all users can access all accounts), you must set up one access group and one data access role and then indicate all users are part of this role. You should also define the access group as the default access group on all of your users (so that new accounts are all labeled with this access group).

Masking Sensitive Data

Refer to Masking Data for instructions describing how to configure the system to mask sensitive data like a customer's social security number or bank account number. If your implementation intends to mask any of the information that appears in the Customer Information Zone please navigate to this zone's documentation for special instructions.

Encrypting Sensitive Data

The product provides support for encrypting certain sensitive data, which may be necessary if your implementation wants to meet security standards such as those for credit card transactions, as specified by the Payment Card Industry Security Standards Council.

Please be sure that you are familiar with the information documented in Database Encryption and Masking before reading further.

The remaining details in this section provide information about the fields in the product that support encryption along with the suggested configuration for the **Encryption** feature configuration entry if your implementation opts to encrypt the data.

External Account ID and Credit Card Number Encryption

If your implementation would like to encrypt credit card IDs, the following entry should be added to the Encryption feature configuration so that the data stored in the External ID column on the Account Automatic Payment table is encrypted:

table='CI_ACCT_APAY', field='EXT_ACCT_ID', encryptedField='ENCR_EXT_ACCT_ID'

NOTE:

- The External Account ID field also exists on other tables such as Auto Pay Clearing Staging table (CI_APAY_CLR_STG) and the Payment Event Upload Staging table (CI_PEVT_DTL_ST). Feature configuration options do not need to be configured for the External Account ID field on these tables as the field on these related tables is encrypted based on the feature configuration specified above for the Account Automatic Payment table (CI_ACCT_APAY).
- A customer's bank account ID or credit card number may also be stored in a column when an order is completed. A field encryption feature configuration for the Order Field table (CI_ENRL_FLD) does not need to be configured if the customer's bank account ID or credit card number is referenced on this table. The encryption for the field should be handled by a Column Reference - Preprocessing algorithm. The base product provides an Encrypt Account Auto Pay External Account Id Column Reference Value algorithm type for this purpose.
- A field encryption feature configuration option needs to be configured for each schema field to be encrypted that represents a customer's bank account ID or credit card number. If the value of an encrypted schema field will later be populated on or compared against an encrypted table field, the schema field and the table field must share the same key alias.

Bank Account Number Encryption

If your implementation would like to encrypt the bank account number stored on the back account table, the following entry should be added to the Encryption feature configuration:

table='CI_BANK_ACCOUNT', field='ACCOUNT_NBR', encryptedField='ENCR_ACCOUNT_NBR'

MICR ID Encryption

The product provides support for encrypting MICR ID associated with payments. In addition, the product supports capturing a hash representation of this field for searching purposes. This field is captured on three tables and entries should be added to the Encryption feature configuration for each as follows:

- **table='CI_PEVT_DTL_ST', field='MICR_ID', encryptedField='ENCR_MICR_ID', hashField='HASH_MICR_ID'**
- **table='CI_PAY_TNDR_ST', field='MICR_ID', encryptedField='ENCR_MICR_ID', hashField='HASH_MICR_ID'**
- **table='CI_PAY_TNDR', field='MICR_ID', encryptedField='ENCR_MICR_ID', hashField='HASH_MICR_ID'**

NOTE: All configurations must share the same key alias.

Person ID Number Encryption

The product provides support for encrypting an identifier associated with a person. In addition, the product supports capturing a hash representation of this field for searching purposes. Because the ID collection on Person includes an ID type, your implementation may decide to encrypt all IDs captured for the person or may choose to only encrypt IDs of one or more specific ID types. If you want to encrypt multiple ID types, you must configure a field encryption option type for each ID type. The following shows an example of an encryption feature configuration entry for encrypting an ID with the type “SSN”:

table='CI_PER_ID', field='PER_ID_NBR', encryptedField='ENCR_PER_ID_NBR', hashField='HASH_PER_ID_NBR', where='ID_TYPE_CD=SSN’

➤

NOTE:

- The Person ID Number also exists on the Order Person ID table (CI_ENRL_PER_ID) table. Feature configuration options do not need to be configured for the Person ID Number field on this table as the field is encrypted based on the feature configuration specified above for the Person Identifier table (CI_PER_ID).
- The Person ID Number may also be stored in a column when an order is completed. A field encryption feature configuration for the Order Field table (CI_ENRL_FLD) does not need to be configured if the Person ID Number is referenced on this table. The encryption for the field should be handled by a Column Reference - Preprocessing algorithm. The base package provides an Encrypt Column Reference Value algorithm type for this purpose.
- A field encryption feature configuration option needs to be configured for each schema field to be encrypted that represents a Person ID Number. If the value of an encrypted schema field will later be populated on or compared against an encrypted table field, the schema field and the table field must share the same key alias.

Encrypting Legacy Data

If you enable encryption in an existing implementation, you should run the encrypt legacy batch controls to encrypt legacy data that persists in the database:

- F1-ENCRT: Encrypt Legacy Table Field Data
- F1-ENCRS: Encrypt Legacy Schema Field Data
- C1-ECRVL: Encrypt Legacy Order Field Col Ref Value

Before executing the batch processes, you must have a keystore file in the system to hold the keys for encrypting data and define an encryption feature configuration that contains the details for the fields you want to encrypt.

Defining Converted COBOL Program Options

The topics in this section describe the transaction that allows you to define the metadata for converted COBOL programs within the current environment's database.

△

CAUTION: Updating converted COBOL Programs 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.

Converted COBOL Program - Main

Use this transaction to define converted COBOL program user exits for your system. Navigate to this page using **Admin** > **System** > **Converted COBOL Program** > **Search.**

Description of Page

The following describes fields that are relevant to defining the user exit code that a converted COBOL Program should use:

Program Component ID represents the internal ID that is given to the converted COBOL program component.

Program Com Name is the physical name of the converted COBOL program component.

Template is the template used to generate the converted COBOL program component.

Location ID *(for development purposes)*

Table *(for development purposes)*

User Exit Program. Specify if you have written user exit code for this converted COBOL program component.

Development Status Flag. Options include Note Generated and Not Regeneratable.

Short Comments provides a short description of the converted COBOL program component.

Long Comments provides more details of the converted COBOL program component.

The **Variable Information** grid (including Variable Name, Occurrence, Variable Sequence, Variable Value, and Variable Alternate Value) is *for development purposes only*.

Advanced Search Options

The product offers an option to perform expanded / fuzzy searching in Control Central Search by Name and Address. The option is only available if your implementation has appropriately configured your database for fuzzy searching.

Refer to the DBA guide in the installation 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.

If your implementation has configured its database to enable fuzzy searching, you must also set up a Feature Configuration to indicate that feature is supported. 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 **Enable Expanded Results** and define a value of **Y**.

Integration

Oracle Utilities Customer to Meter a provides tools to facilitate the integration other systems. This section provides technical information needed by your implementers to accomplish this.

CTI-IVR Integration

Oracle Utilities Customer Care and Billing provides tools to facilitate the integration with your Computer Telephony Integration/Interactive Voice Response (CTI/IVR) system. The interface provides the following functionality:

- The ability to launch Control Central for a particular account ID or phone number from an external application
- The ability to accept the next call, as dictated by the CTI software,
- The ability to perform an outbound phone call from within Oracle Utilities Customer Care and Billing

This document provides technical information needed by your implementers to fully integrate with your CTI/IVR system.

Launching The System From an External Application

The following section describes possible options to launch the system from an external system.

Launching The Application Using a URL

You launch the application using a URL. With this option you can set the system to launch a script upon startup. You can also indicate to the system to automatically load an appropriate page (if this information is not part of the script).

The application includes a simple sample html page that can launch CCB Control Central Search by Phone Number or by Central Search Account Id. The page is called CTISample.HTM and is located in /cm_templates.

FASTPATH:
Refer to Launching A Script When Starting The System for further information.

Receiving the Next Caller in the Queue

If your CTI-IVR system allows users to request the next caller waiting in a queue, the system provides a mechanism to integrate with this functionality.

A BPA script called **Get Next Caller (C1-CCByAcct)** is available that can be used to request the next call waiting in an inbound queue managed by a CTI application.

When the **Get Next Caller** BPA is executed, it launches a browser script function called **launchCTI** located in a file called **ext_cti.jsp**. The **launchCTI** function calls a function called **ctiGetNextCaller** to retrieve the next caller's account ID and uses Launching A Script When Starting The System to load Control Central for an account.

Customize Integration to Your Next Caller Function

The ext_cti.jsp file shipped with the base product provides sample functionality that should be replaced with the appropriate integration to your CTI application. In the sample provided, the **ctiGetNextCaller** randomly takes an account ID from a predefined list of accounts.

In order to integrate the next caller functionality with your CTI-IVR system, perform the following steps:

- Copy the JSP page ext_cti.jsp from the /cm_templates directory found under the web application root directory on your Oracle Utilities Customer Care and Billing server to the /cm directory.
- In the /cm directory, replace the contents of the **ctiGetNextCaller** function to retrieve the next caller ID from your CTI application.

Initiating an External Call

This section describes the automated dialer functionality provided with the system as well as information about integrating with your own automated dialer.

Overview of Automated Dialer

In order to initiate a call to a customer from within the system, a context menu item **Go To Automated Dialer** is available on the Person context menu. To call a customer displayed in the current context, choose this option from the person context menu and a window appears, showing a list of phone numbers defined for that person.

Select the desired phone number and click **Dial**.

NOTE:
Context Entry Secured. The navigation key for this window automatedDialer refers to an application service to facilitate application security. If your installation does not support an integration with external dialer software, configure the security settings to ensure that users do not have access to the application service for this context entry.

Technical Implementation of Automated Dialer

The popup window is implemented as a JSP page, which calls the JSP page ext_cti_dialer.jsp to integrate with an automated dialer. The ext_cti_dialer.jsp page provided with the system integrates to any soft phone protocol handler that launches a dialer based on overriding the "tel:" protocol from a browser, for example, Cisco IP Communicator.

Phone Dialer Configuration

If your implementation chooses to use the functionality provided with the system and integrate with a soft-phone dialer, you must copy the JSP page ext_cti_dialer.jsp from the /cm_templates directory found under the web application root directory on your Oracle Utilities Customer Care and Billing server to the /cm directory:

Customize Integration to Your Automated Dialer Software

In order to integrate with a different automated dialer software application, your implementers must modify the ext_cti_dialer.jsp to call the appropriate dialer.

- Copy the JSP page ext_cti_dialer.jsp from the /cm_templates directory found under the web application root directory on your Oracle Utilities Customer Care and Billing server to the /cm directory.
- Make the appropriate changes to the copy of ext_cti_dialer.jsp in the /cm directory to integrate with your automated dialer (e.g. change the protocol from “tel:” to “callto:”.

Customize Automated Dialer User Interface

Your implementation may choose to display a different user interface for the **Go To Automated Dialer** function than the one provided with the system. For example, perhaps there is more information that you would like to display in addition to the person's name and phone numbers. In order to do this, perform the following steps:

- Create your customized component to provide the desired functionality.
- Create a navigation key for your new component and indicate the URL being overridden. The remainder of the section walks you through these steps.

Go to **Utilities, System, Navigation Key +**.

For **Navigation Key,** specify a name for the new navigation key prefixed with CM.

For **URL Location**, select **External (Override)** to override a base navigation key.

When you select **External (Override),** the **Overridden Navigation Key** becomes available. Select the **automatedDialer** navigation key because that is the key you are overriding.

The **URL Override** is the path on the web server to your custom component.

When overriding a navigation key, you must flush the system login cache on the web server. The navigation keys are stored in the system login cache, so the overrides do not become effective until the cache is flushed. To flush the cache, issue the following command in your browser's address bar: http://server:port/flushSystemLoginInfo.jsp , where server is the name or address of your web server and port is the port number of the application, for example, http://CD-Implementation:7500/flushSystemLoginInfo.jsp.

FASTPATH:
Refer to the Defining Navigation Keys for more information.

Analytics Integration Overview

Oracle Utilities Customer to Meter provides tools to facilitate the integration with Oracle Utilities Analytics. \

Analytics Configuration

This section describes functionality provided for integrating with Oracle Utilities Analytics (OUA).

- **Customer Analytics Configuration Portal**

Oracle Utilities Customer to Meter provides a Customer Analytics Configuration portal that holds information on the customer analytics configuration activities. This portal provides a summary of how much configuration has been set up, and also provides links and guidelines for the areas that need configuration, at the minimum, to successfully extract data from Oracle Utilities Customer Care and Billing to OUA. The following provides more information about this functionality.

1. **Customer Analytics Configuration Details** – These data load parameters are used to extract, load, and transform data from Oracle Utilities Customer Care and Billing to Oracle Utilities Analytics. These parameters are used to identify and/or filter data when loading them into the data warehouse. Refer to the *Oracle Utilities Customer Care and Billing* section in the ***Configuring Oracle Utilities Analytics*** chapter of the *Oracle Utilities Analytics Administration Guide* for detailed information on each extract parameter.

2. **Bucket Configuration List** – This is a summary of the bucket configurations that are needed to classify and group extracted data within Oracle Utilities Analytics. The bucket configurations provide the ability to readily report and analyze data by buckets, or groups. Refer to the *Oracle Utilities Customer Care and Billing* section in the ***Configuring Oracle Utilities Analytics*** chapter of the *Oracle Utilities Analytics Administration Guide* for detailed information on each bucket configuration supported by Oracle Utilities Customer Care and Billing.

- **Characteristic Mapping**

  Oracle Utilities Customer Care and Billing has identified the objects for which characteristic extraction is supported. This includes mapping the source characteristic entity to thetarget dimension, and defining which target columns are allowed to be mapped for which target dimension. The following table shows the supported mapping.

Object	Supported Target Dimension	Supported Target Columns
Account Characteristic	Account Dimension	User Defined Fields 6 - 15
Person Characteristic	Person Dimension	User Defined Fields 1 - 6
Premise Characteristic	Premise Dimension	User Defined Fields 6 - 18
Premise Characteristic	Address Dimension	User Defined Fields 7 - 16
Service Agreement Characteristic	Service Agreement Dimension	User Defined Fields 9 - 19

Meter Analytics Configuration Portal

The Meter Analytics Configuration portal holds information on the meter data analytics configuration activities. This portal provides a summary of how much configuration has been set up, and also provides links and guidelines for the areas that need configuration, at the minimum, to successfully extract data from Oracle Utilities Meter Data Management to OUA. The following provides more information about this functionality.

- **Analytics - Oriented Master Configuration:** These data load parameters are used to extract, load, and transform data from Oracle Utilities Meter Data Management to Oracle Utilities Analytics. They are used to identify and/or filter data when loading them into the data warehouse. Refer to the *Oracle Utilities Meter Data Management* section in the ***Configuring Oracle Utilities Analytics*** chapter of the *Oracle Utilities Analytics Administration Guide* for detailed information on each extract parameter
- **Analytics - Oriented Extendable Lookup List:** The meter data analytics related Extendable Lookup Business Objects are listed in this section. It provides a navigation link to the extendable Lookup Maintenance portal where the lookup values can be configured. Refer to the *Oracle Utilities Meter Data Management* section in the ***Configuring Oracle Utilities Analytics*** chapter of the *Oracle Utilities Analytics Administration Guide* for detailed information on meter data analytics related extendable lookups.
- **Service Point Configuration List:** This lists the service point types within the system and indicates for each whether or not the meter analytics configuration has been setup. It provides a navigation link to the service point type where the necessary configuration can be setup or modified. Refer to the *Oracle Utilities Meter Data Management* section in the ***Configuring Oracle Utilities Analytics*** chapter of the *Oracle Utilities Analytics Administration Guide* for further information.

Batch Controls

Batch Code	Name / Description
D1-SPSDL	Service Point Snapshot Download
D2-DUGDL	Usage Subscription Usage Group Download
D2-MVREF	Materialized View Refresh
D2-USGDL	Service Point Usage Snapshot Download
D2-UUSDL	SP Unreported Usage Snapshot Download
D2-VEEDL	SP VEE Exception Snapshot Download

References

Refer to the following documentation for more information about Oracle Utilities Analytics:

- *Oracle Utilities Analytics Admin Guide*
- *Oracle Utilities Analytics Dashboards for Oracle Utilities Customer Analytics, Revenue Analytics and Credit & Collections Analytics Metric Reference Guide*
- *Oracle Utilities Analytics Dashboards for Oracle Utilities Exception Analytics Metric Reference Guide*
- *Oracle Utilities Analytics Dashboards for Oracle Utilities Meter Data Analytics Metric Reference Guide*
- *Oracle Utilities Extractors and Schema for Oracle Utilities Customer Care and Billing Data Mapping Guide*
- *Oracle Utilities Extractors and Schema for Oracle Utilities Meter Data Management Data Mapping Guide*

Self-Service Integration

The following sections describe functionality provided for the integration with self-service applications such as Oracle Utilities Customer Self Service (OUCSS).

About Self-Service Tasks

Self-service tasks hold information about self-service operations that customers initiate from an integrated self-service application, such as Oracle Utilities Customer Self-Service (OUCSS).

A few examples of self-service tasks are:

- One time payments
- Account verification
- Start/Stop requests
- Auto pay setup
- Creating customer meter readings

Refer to the Oracle Utilities Customer Self-Service (OUCSS) documentation for more details on supported self-service operations.

The following sections highlight self-service integration functionality.

Self-Service Task Type Defines Self-Service Task Behavior

Each type of self-service operation has specific processing requirements.

Self-service task type controls how a self-service task is processed. It defines the business object to use when creating the self-service task record. The business object defines the lifecycle of self-service task record.

The self-service task type can also specify common configuration data, such as settings for error handling.

Creating Self-Service Tasks

The customer self-service application invokes inbound web services to create self-service tasks. The system supplies an inbound web service for each supported self-service operation.

Refer to the Oracle Utilities Customer Self-Service (OUCSS) documentation for more details on supplied inbound web services.

Each inbound web service references a service script that processes the specific self-service operation. Refer to the service script system data for processing details.

Setting Up Self-Service Task Configuration

Self-Service Integration Configuration

The integration with a self-service application such as Oracle Utilities Customer Self Service (OUCSS) requires the setup of master configuration data that controls the processing of your self-service operations.

To set up self-service integration master configuration, navigate using **Admin** > **General** > **Master Configuration.**Two master configurations are required:

- **CCB Self Service Integration:** This master configuration relates to the integration of customer data with a self service application.
- **MDM Self Service Master Configuration:** This master configuration is used for the integration of meter data with a self service application.

Refer to the embedded help for more details on how each section in these records is configured.

Defining Self-Service Task Types

A self-service task type defines properties that control how a self-service task is processed.

Refer to About Self-Service Tasks for an overview of service task functionality.

To maintain the types applicable to your product, open **Menu** > **Self-Service** > **Self-Service Task Type.**

This is a standard All-in-One portal.

The information captured on the self-service task type depends on the business objects supported by your product. Refer to the embedded help text for more information.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference F1_SVC_TASK_TYPE.

Maintaining Self-Service Tasks

This section describes the functionality supported for viewing and maintaining self-service tasks.

Refer to About Self-Service Tasks for an overview of self-service task functionality.

Navigate using **Menu** > **Self-Service** > **Self-Service Task** > **Search**. You are brought to a query portal with options for searching for self-service tasks.

Once a self-service task record has been selected, you are brought to the maintenance portal to view and maintain the selected record.

The **Self-Service Task** zone provides basic information about a self-service task. Refer to the embedded help for more information.

Configuring Rate Compare Usage Adjustment Profiles

This section describes the steps involved in setting up usage adjustment profiles for use with the Rate Compare feature in Oracle Utilities Customer Self- Service.

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).
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 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 CCB Rate Schedule extendable lookup to associate the usage group that contains the "Profile Accumulation" usage rule to a rate schedule.
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 MDM 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 CCB 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
- 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

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**

Profile Accumulation Usage Group and Rule

Create a usage group that contains a "Profile Accumulation" usage 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 Rule

- **Usage Group: Electric Residential Interval KWH**
- **Usage 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

CC&B Rate Schedule Extendable Lookup

Create an entry in the "CCB Rate Schedule" extendable lookup to associate the usage group that contains the "Profile Accumulation" usage rule to an applicable rate schedule in Oracle Utilities Customer Care and Billing.

Example CCB Rate Schedule extendable lookup:

- **Rate**: E-INT-RES
- **Description**: Electric Residential Interval Rate
- **Default Usage Group**: Electric Residential Interval KWH (E-INT-RES)

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**

MDM Self-Service Master Configuration - Rate Compare Configuration

Configure the "Rate Compare Configuration" section of the MDM 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 CCB Rate Schedule extendable lookup).

Example MDM 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

Defining DataConnect Options

Oracle Utilities DataConnect facilitates data extraction from Oracle Utilities Customer to Meter 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.

DataConnect Data Extracts

Oracle Utilities DataConnect can be used to extract three types of data:

- **Master Data**

  (see Master Data Extracts *for more information)*

  - Service agreement, account, and person related information
  - Service point related information
  - Install event related information

- **Billing Data**

  (see Billing Data Extracts *for more information)*

- **Measurement Data**

  (see Consumption Data Extracts *for more information)*

The data extracted can be correlated in external systems using the service agreement and/or service point data that is included in each type of extract. All data extracts contain a combination of the following data elements that can be used for this correlation:

- **CCB Service Point ID**: The service point ID in Oracle Utilities Customer to Meter and/or
- **MDM Service Point ID**: The service point ID in Oracle Utilities Meter Data Management and/or
- **Service Agreement ID:** The service agreement ID in Oracle Utilities Customer to Meter

External systems receiving data extracted from the Oracle Utilities DataConnect facilities can use these IDs to associate billing data and measurement data to extracted master data.

Master Data Extracts

Master data extracts are performed through the use of data synchronization, audit algorithms, business objects, and related batch processes.

Batch processes are used to create initial load extracts for service agreement / account / person, service point, and install event related master data. The initial load includes the historical data in the system (either all or a subset, depending on how the batch process is configured). Following the initial load, data synchronization requests are created when master data is added or changed.

Master Data Extracts: Maintenance Object Audit Algorithms

Detecting changes in master data can be identified by audit algorithms on the following maintenance objects:

- Service Agreement
- Account
- Person
- Service Point (MDM Service Point)
- Install Event
- Device

The Generic Change Data Capture algorithm type (F1-GCHG-CDCP), which is provided in the base package, can be used to create audit algorithms for following maintenance objects:

- Service Agreement
- Service Point (MDM Service Point)
- Install Event

In addition,

- An **Account Change Data Capture (SA-Based) – Include Reactivated SAs** algorithm type (C1-ACCCDCSAR) is provided in the base package. This algorithm type can be used to create an audit algorithm on the Account maintenance object to determine if a service agreement sync request record is to be created for each of the account's non-closed, non-canceled service agreement(s). A change in a related account's details will instantiate a Service Agreement Extract Sync Request (of the type defined for the SA Sync Request BO algorithm parameter) if one does not already exist in the initial state for the service agreement. The base package also contains an algorithm (C1-ACCECDCSA) that is based on this algorithm type.
- A **Person Change Data Capture (SA-Based) – Include Reactivated SAs** algorithm type (C1-ACCCDCSAR) is provided in the base package. This algorithm type can be used to create an audit algorithm on the Person maintenance object to determine if a service agreement sync request record is to be created for each of the person's non-closed, non-canceled service agreement(s). A change in the main person details linked to the service agreement's account will instantiate a Service Agreement Extract Sync Request (of the type defined for the SA Sync Request BO algorithm parameter) if one does not already exist in the initial state for the service agreement. The base package also contains an algorithm (C1-PERECDCSA) that is based on this algorithm type.
- A **Device Change Data Capture (Install Event-Based)** algorithm type (D1-IEDV-CDCP) is provided in the base package. This algorithm type 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.

Master Data Extracts: Business Objects and Algorithms

The maintenance object audit algorithms create data synchronization requests in the following manner:

- If the Generic Change Data Capture (F1-GCHG-CDCP) algorithm type (provided in the base package) is used, the synchronization requests are based on the Sync Request BO maintenance object options.
- Alternatively, if the base package supplied Maintenance Object – Audit algorithm types, which are outlined in the Master Data Extracts: Maintenance Object Audit Algorithms section are used, the synchronization requests are based on the Sync Request BO defined in the algorithm parameters.

Extraction of service agreements, service points and install events related information is supported by the following data synchronization business objects.

- **SA/Account/Person Sync for DataConnect (C1-ExternalRepoSASync):** used to extract service agreement / account / person related information. This business object should be defined as a value for the Sync Request BO option on the Service Agreement maintenance object.
- **SP Sync for 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 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:

- Capture SA-Based Initial Snapshot for DataConnect (C1-CEXTRCSAI)

- Capture SP Initial Snapshot for DataConnect (D1-SPEINISNP)
- Capture Install Event Initial Snapshot for DataConnect (D1-IEEINISNP)

These algorithms specify the batch control used for the extract process (see below).

The Sync Request Monitor batch control (F1-SYNRQ) monitors synchronization requests in the Pending state and executes Monitor algorithms that check for other pending related synchronization requests, and, if none, transitions them to the Determine if Sync Needed state.

Enter algorithms on the Determine If Sync Needed states verify if a synchronization is needed by capturing the final snapshot of the data and comparing it against the initial snapshot. If changes are detected, the final snapshots of the data are stamped on the synchronization requests for export.

- Capture SA-Based Final Snapshot for DataConnect (C1-CEXTRCSAF)
- Capture SP Final Snapshot for DataConnect (D1-SPEFNISNP)
- Capture Install Event Final Snapshot for DataConnect (D1-IEEFNISNP)

The Prepare Delimited Extract Data (Ignore Custom Sync Records) Enter algorithm (C1-PRPEXDNCU) or the Prepare Delimited Extract Data Enter algorithm (D1-PRPEXTDTA)on the Send Request state prepares and formats 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).

Master Data Extracts: 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:

- **SA-Based Initial Load for DataConnect (C1-SAEIL):** This batch control creates initial synchronization requests for service agreements / accounts / persons.
- **SP Initial Load for DataConnect (D1-SPEIL):** This batch control creates initial synchronization requests for service points.
- **Install Event Initial Load for DataConnect (D1-IEEIL):** This batch control creates initial synchronization requests for install events

The following batch processes are used to create extract files from synchronization requests.

> ➤ **NOTE:** These batch processes should be run to create initial load data synchronization requests based on the Sync for DataConnect business objects.

- **SA-Based Extract for DataConnect (C1-SASYX):** This batch control creates extract file(s) for service agreements / accounts / persons related information.
- **SP Extract for DataConnect (D1-SPESR):** This batch control creates extract file(s) that contain service point information.
- **Install Event Extract for 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 this file.) for a delimited flat file containing extracted data.

Setting Up and Executing Master Data Extracts

Setting up master data extracts involves the following steps:

1. Add audit algorithms on the Service Agreement, Account, Person, MDM Service Point, Install Event, and Device maintenance objects.
    a) Add the Generic Change Data Capture algorithm as an audit algorithm on the Service Agreement, MDM Service Point, and Install Event maintenance objects.
    b) Add the Account Change Data Capture (SA-Based) - Include Reactivated SAs algorithm as an audit algorithm on the Account maintenance object.
    c) Add the Person Change Data Capture (SA-Based) - Include Reactivated SAs algorithm as an audit algorithm on the Person maintenance object.
    d) Add the Device Change Data Capture (Install Event-Based) algorithm as an audit algorithm on the Device maintenance object.
2. Add the DataConnect synchronization request business objects as Sync Request BO options on the Service Agreement, Service Point and Install Event maintenance objects.
3. Execute initial load batch processes related to service agreements, service points and install events.
4. Once the initial load synchronization has been executed, changes to service agreements, accounts, persons, service points, install events or related devices will trigger the creation of new synchronization requests and resulting extraction files.
5. Execute the sync request monitor process.
6. Execute the extract batch processes related to service agreements, service points and install events.

Billing Data Extracts

When sending billing related 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 includes data history for all bill segments for a specified historical period that are:

- Linked to completed bills.
- Not linked to bills but are linked to frozen FTs that are linked to completed bills.

Current data should be extracted on a regular ongoing (or incremental) basis. However, in addition to sending current data, any billing corrections should be extracted as well.

Extraction of billing data is performed through use of data synchronization, post bill completion algorithms, business objects and related batch processes.

Billing Data Extracts: Business Objects and Algorithms

Billing generates synchronization requests when bills complete through a Customer Class – Post Bill Completion algorithm. The Change Data Capture of Billing Data for DataConnect (C1-CAPBILLEX) algorithm type provided in the base package can be used to create a Customer Class – Post Bill Completion algorithm that creates a billing synchronization request record for each of the bill’s bill segments if one does not exist in the initial state.

Billing related information extracts are supported by the following data synchronization business object.

- Billing Sync for DataConnect (C1-ExternalRepoBillingSync). This business object should be defined as an algorithm parameter on the algorithm based on the Change Data Capture of Billing Data for DataConnect (C1-CAPBILLEX) algorithm type.

This business object use the following Pre-Processing algorithm to take an initial data snapshot, and define the batch control used to extract data and export to a flat file. However, since billing data are always synched when it gets completed, no actual initial snapshot is taken. Instead, the parameters used for building the snapshots are taken and stored on the sync request record for later use when building the final snapshot.

- Capture Billing Data Initial Snapshot for DataConnect (C1-EXTRCBLIP)

The Sync Request Monitor batch control (F1-SYNRQ) monitors synchronization requests in the Pending state and executes Monitor algorithms that check for pending related synchronization requests, and, if none, 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 if the billing related data is to be extracted and exported.

- Determine if the Billing Sync is Needed (C1-DSCANUSNS) - this is the algorithm that checks if there is a need to export the billing data. It will discard the synchronization request if one of the following is true:
  - The synchronization request is for a cancelled bill segment whose original usage has not been previously sent out.
  - The synchronization request has been marked for non-synchronization (such as, the bill segments related SA Type is not configured to be extracted on the Billing Data Configuration for DataConnect master configuration record).
- Capture Billing Data Final Snapshot for DataConnect (C1-CAPBILUSN) - this is the algorithm that captures the billing data information and stores it on the synchronization request.

The Prepare Delimited Extract Data (Process All) Enter algorithm (C1-PRPEXDCUS) on the Send Request state prepares and formats the data for extraction, and creates a general process record for the synchronization request (based on the extract batch control defined by the pre-processing algorithm).

Billing Data Extracts: Batch Controls

The billing data extraction process uses a set of batch processes to create data synchronization requests and extract files.

The following batch process is used to create initial synchronization requests:

- Billing Data Initial Load for DataConnect (C1-BLEIL): This batch control creates initial synchronization requests for bills.

This batch processes should be run to create initial load data synchronization requests based on the C1-ExternalRepoBillingSync business object.

The following batch process is used to create extract files from synchronization requests.

- Billing Data Extract for DataConnect (C1-BSYEX): This batch control creates extract file(s) for billing related information.

This batch control is defined as a value for the Batch Control for Extract algorithm parameter on the Pre-Processing algorithm on the C1-ExternalRepoBillingSync business object.

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 this file) for a delimited flat file containing extracted data.

Setting Up and Executing Billing Data Extracts

Setting up billing data extracts involves the following steps:

1. Configure the Billing Data Configuration for the DataConnect master configuration record. This is used to configure the UOM, TOU, and SQI of the SA Type's service quantity and charges that will be extracted. Additionally, you can also configure the UOM, TOU and SQI at the rate level.
2. Add the Change Data Capture of Billing Data for DataConnect (C1-CAPBILLEX) related post bill completion algorithm to the applicable customer classes.
3. Execute initial load batch processes related to billing data.
4. Once the initial load synchronization has been executed, current data should be extracted on a regular ongoing (or incremental) basis. In addition to sending current data, any billing corrections will be extracted as well (data related to bill segments not linked to bills but are linked to frozen FTs that are linked to completed bills).
5. Execute the sync request monitor process.
6. Execute the extract batch processes related to billing.

Billing Data Configuration for DataConnect Master Configuration

The master configuration business object for Billing Data Configuration for DataConnect defines general configuration details for the DataConnect functionality. This is used to configure the UOM, TOU, and SQI of the SA Type's service quantity and charges that will be extracted. Additionally, you can also configure the UOM, TOU and SQI at the rate level.

Navigate using **Admin** > **General** > **Master Configuration.** In the Master Configuration list, scroll to Billing Data Configuration for DataConnect and click the add icon if there is no record yet or the edit icon to modify an existing record.

For more information about specific fields in the master configuration, refer to the embedded help.

Consumption Data Extracts

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.

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 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:

- **Consumption Extract Type (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.
- **Consumption Extract Type 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 general process record will be written 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 general process record 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 a general process record 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.
- The "Create General Process Record for Re-derived Values" algorithm creates general process records 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 for DataConnect (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 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

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 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.

Consumption Extract Requests - 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 and file name for this file.) for a delimited flat file containing extracted data.

Consumption Extract Requests - Setup Steps

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.
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:
    1. 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.
    2. The **Request Monitor (Deferred)** (F1-SUBRQ) batch control should be used to monitor for pending requests and transition them to the "Submit Job" state.
    3. **Historical corrections consumption extract batch controls** (based on D2-IDEHC, one for each consumption extract type), should be configured to run on a regular basis to check for and create historical correction extracts.

Extract Flat File Formats

This section outlines the format of output flat files created by master and billing data extracts.

Data Areas:

- C1-ExternalRepoSABasedSnapshot
- D1-ExternalRepositorySPSnapsht
- D1-ExternalRepositoryIESnapsht
- C1-ExternalRepoBillingSnapshot
- D2-IntervalDataExtRepoSnapshot (used for interval consumption extract)
- D2-IntervalDataExtRepoTOUSnap (used for interval consumption extract mapped to TOU periods)

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.

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
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
```

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. See Business Flags for more information.

This section provides further information about business flags and their use in the system.

Understanding Business Flags

Refer to About Business Flags in the *Oracle Utilities Framework Administration User Guide* for detailed information about the components of business flags.

Customer to Meter allows for business flags that typically represent situations, exist at a customer's service point, and are shared across external applications; such as, Oracle DataRaker, Oracle Utilities Mobile Workforce Management / Oracle Field Service Cloud. and Oracle Utilities Operational Device Management. The situations that a business flag can represent range from critical knowledge that requires manual analysis for resolution to purely informational notifications.

Sharing information allows users of each system to quickly understand the status of a service point (or device, or network location in some cases) as it relates to situations or insights that would benefit users of the system. For example, the advanced analytics system (Oracle DataRaker) may identify potential theft situations that should be investigated prior to releasing the next bills from CCB with usage transactions from MDM for the related service point’s consumption.

The subsequent sections describe business flag functionality that is specific to the service point business flag, as implemented by this product.

When creating a new MDM device type there are the following options:

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

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.

Business Flag Available Actions

Allowed actions are implemented as BPA scripts 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 activity will be selected by the user.
- **Create Case:** This will initiate a case associated to the business flag's service point. The type of case will be selected by the user.
- **Create Notification:** This will initiate communication with the customer for the business flag's service point. The type of communication will be determined by the notification type selected by the user.
- **Create Service Issue Monitor:** This will initiate a Service Issue Monitor (SIM) process associated to the business flag's service point. The SIM type is identified on the Business Flag Type configuration. If there are already existing SIM records of this type for the service point, the most recent SIM process will be linked to the business flag.

Business Flag Adding Available Actions

To extend the actions that are available to the user for investigating a business flag the following steps can be taken:

1. A new BPA script should be developed that provides the functionality for your particular business use case.
2. That BPA script will be added to the transactional business object for the business flag using the option type “Valid Action”.
3. Edit the business flag type to include the newly created action in the “Allowed Actions” section of the business flag type.

Making Allowed Actions Available

Allowed actions are only accessible on those business object statuses that have been configured to support them. To permit access to the available actions on a given status one must add a business object status option for the type “Execution of Valid Actions Permitted” with a value of “Y”.

Business Flag Can Prevent Bills from Freezing

Some business flags, particularly those that require investigation, should place a hold on the billing process to prevent potentially incorrect bills from being issued to the customer. When certain types of business flags are present on a service point it indicates that there may be a problem with the reported usage that is underlying the bill and as such it may not be desirable to allow the billing process to complete until the investigation into that usage has been completed. For example, if the business flag indicated potential theft or abnormally high usage it may be preferable to delay issuing the bill until an investigation can be performed and the issue rejected or resolved.

This functionality can be enabled through configuration of the Prevent Bill Segment Freeze Due to Business Flag algorithm.

See algorithm type Prevent Bill Segment Freeze Due to Business Flag (C1-PRVBSFBF) for more detail.

In MDM, Usage Transactions can also be prevented by configuring the *Valid Business Processes for Hold* with *Usage Processing* on the Service Point Business Flag Type.

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 Business Flag Integration Configuration in the Oracle Utilities Application Framework *Administrative User Guide*.

Business Flags Can Require Analysis

For more serious situations a business flag can require manual analysis from a user. These types of business flags will generate a To Do when they are received from the external application. To resolve the To Do a user will investigate the service point based on the type of business flag present. Generally this investigation will result in the business flag being either confirmed or rejected. To reach the conclusion of whether the business flag should be confirmed or denied the user can be provided with a set of available actions that can be performed on the business flag.

See C1-BFANL for more information.

Configuring Business Flag Integration

Oracle Utilities Meter Data Management includes the following components used when integrating with Oracle DataRaker:

Master Configuration

The following master configurations are used to configure the sync process between MDM and Oracle DataRaker:

Master Configuration	Description
DataRaker Integration Master Configuration	This master configuration is used to enable deep linking into Oracle DataRaker. 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 the Oracle DataRaker point types.

Service Provider Processing Methods

The following processing methods should be associated to the service provider configured for Oracle DataRaker:

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 reviewed for Oracle DataRaker:

Extendable Lookup	Description
Business Flag Standard Name	If Oracle DataRaker 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 Oracle DataRaker.

A Field Activity 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 drive the resolution 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.

Pease refer to algorithm type Business Flag Confidence Update (C1-BSFLGCFUP) for more detail.

Business Flags Can Create Alerts

Some business flags have a high enough importance that they should be highlighted to any user that is viewing data for the service point the business flag is related to. Whether a business flag should create an alert is controlled on the business flag type. Furthermore, the appropriate algorithm must be configured in the “Installation Options – Framework” “Algorithms” page for the “Control Central Alert” system event.

The product delivers the algorithm type “Highlight Open Business Flags”

Pease refer to algorithm type *Business Flag Confidence Update (* C1-BSFLGCFUP*)* for more detail.

Customer Relationship Management Integration

Oracle Utilities Customer Care and Billing provides tools to facilitate the integration with customer relationship management applications such as Oracle Social Relationship Management (SRM).

The following sections describe this functionality.

About Customer Relationship Requests

A customer relationship request is used to process requests that are initiated from an external customer relationship management system.

The following sections highlight customer relationship request functionality.

Social Relationship Management Requests

Oracle Social Relationship Management (SRM) is an example of an external system that can initiate customer relationship requests.

Social network posts cover a vast range of topics, a number of which could be related to the users' concerns with their utility service. For example:

- A Facebook user could post on a community page about a power outage in a specific area.
- A Twitter user might post about specific problems with his/her utility service — e.g. high bill complaints.

Social conversations of this nature can be monitored and managed via Oracle SRM. Oracle SRM's Engage module allows its users to see all social conversations, identify customer-specific issues and route the messages to the appropriate systems/ channels.

Requests from social relationship management systems typically include:

- Information about the original posting on the social network/community — e.g. author's identifier, link to the post, etc.
- Information about the posting within the social relationship management system — e.g. the user that sent the request to the utility, link to the post, etc.
- Additional details that will help with processing/resolution — e.g. customer/account numbers, address, bill IDs, etc.

Creating Customer Relationship Requests

Most customer relationship requests will be initiated from an external system.

The base product provides an Add Customer Relationship Request (C1–AddCustRR) inbound web service that can be used in a customer relationship management integration to create customer relationship request records. This inbound web service invokes a processing script that is configured in Customer Relationship Integration master configuration. The base product supplies an example of this processing script. It maps the inbound service's input data into the customer relationship request record. Refer to the C1-CreCustRR service script system data for processing details.

Setting Up Customer Relationship Request Configuration

The following topics highlight the general configuration steps required to use customer relationship request functionality and to integrate with a customer relationship management application such as Oracle Social Relationship Management (SRM).

Customer Relationship Management Integration Configuration

The integration with a customer relationship management application such as Oracle Social Relationship Management (SRM) requires the setup of master configuration data that controls the processing of your customer relationship requests.

To set up customer relationship integration master configuration, navigate using **Admin** > **General** > **Master Configuration**. You are brought to an all-in-one portal with options to add/edit the Customer Relationship Integration master configuration record. Refer to the embedded help for more details on how each section in this record is configured.

Defining Customer Relationship Request Types

A customer relationship request type defines properties that control how a customer relationship request is processed.

Refer to About Customer Relationship Requests for an overview of customer relationship request functionality.

To maintain customer relationship request types, open **Admin** > **Integration** > **Customer Relationship Request Type** > **Search**.

This is a standard All-in-One portal.

The information captured on the customer relationship request type depends on the business objects supported by your implementation. Refer to the embedded help text for more information.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference C1_CUST_REL_REQ_TYPE.

Maintaining Customer Relationship Requests

This section describes the functionality supported for viewing and maintaining customer relationship requests.

Refer to About Customer Relationship Requests for an overview of customer relationship request functionality.

Navigate using **Menu** > **Integration** > **Customer Relationship Request** > **Search**. You are brought to a query portal with options for searching for customer relationship requests.

Once a customer relationship request record has been selected, you are brought to the maintenance portal to view and maintain the selected record.

The **Customer Relationship Request** zone provides basic information about a customer relationship request. Refer to the embedded help for more information.

The **Notes** zone allows users to capture additional notes to the customer relationship request.

RightNow Knowledge Integration

Oracle Utilities Customer Care and Billing provides an integration with Oracle RightNow Knowledge that allows users to view frequently asked questions about business processes/functions that relate to the portal or page that they are currently viewing and/or working on.

The following sections describe this functionality.

Displaying Frequently Asked Questions

The Oracle RightNow Knowledge application can be used to develop content for frequently asked questions about business processes. It also functions within the Oracle Customer Care and Billing application. The RightNow KnowledgeSyndication widget defines the interface for the retrieval of this content.

The Oracle Customer Care and Billing application provides a **Frequently Asked Questions** dashboard zone that uses the RightNow KnowledgeSyndication widget to retrieve content from a RightNow Knowledge base instance. This zone is context-sensitive.

Configuration data controls how information is displayed in the **Frequently Asked Questions** dashboard zone. Refer to RightNow Knowledge Integration Configuration for more details.

RightNow Knowledge Integration Configuration

The integration with Oracle RightNow Knowledge requires the setup of master configuration data that controls how content from the RightNow Knowledge base is retrieved and presented on the **Frequently Asked Questions** dashboard zone. This configuration includes the following settings:

- The URL of the RightNow Knowledge base instance.
- The list of portals and pages for which the **Frequently Asked Questions** dashboard zone will display.
- Settings for each portal / page:
  - Search terms to pass to the RightNow KnowledgeSyndication widget. The base product supplies a number of common search term types. Implementation-specific search term types can be added.
  - Number of results/answers to display in the **Frequently Asked Questions** dashboard zone.

To set up the RightNow Knowledge Integration master configuration, navigate using **Admin** > **General** > **Master Configuration.** You are brought to an all-in-one portal with options to add/edit the RightNow Knowledge Integration master configuration record. Refer to the embedded help for more details on how each section in this record is configured.

External Applications

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, outage management systems, 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. .

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.

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** > **Integration** > **External Applications**.

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.
- **Service Provider:** 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.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference D1_SPR.

Configuring Master Data Synchronization

Refer to Customer To Meter Master Data Synchronization for more information.

Oracle Utilities Operational Device Management

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) and device installation information (install events)
- Oracle Utilities Operational Device Management is typically considered the "system of record" for assets (devices)

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 are created or changed in ODM, corresponding devices must be created or changed in MDM
- **Service Point/Contact - Asset Location/Contact**: As Service Points and/or Operational Contacts for Service Points 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 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. 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 in Oracle Utilities Mater Data Management based on assets 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 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

Oracle Dataraker

Overview

This section provides an overview of how Oracle Utilities Meter Data Management supports integrations with Oracle DataRaker. In an integration between Oracle Utilities Meter Data Management and Oracle DataRaker:

- Either Oracle Utilities Meter Data Management or Oracle DataRaker can initiate a business flag
- Oracle Utilities Meter Data Management can notify Oracle DataRaker of a change in status of a business flag:
- A business flag in error has been discarded
- A business flag's confidence has been updated
- The results of field work initiated by a business flag

For additional details see About Business Flags in the Oracle Utilities Application Framework *Administrative User Guide*.

Oracle Utilities Network Management System

Outage Care Integration

The following section describes functionality provided for the integration between Oracle Utilities Customer Care and Billing and Oracle Utilities Network Management System.

The Big Picture of Outage Care Integration

Oracle Utilities Customer Care and Billing is the central repository for customer information; for example, name, address, phone number, etc. Oracle Utilities Network Management System is the central repository for outage information; for example, outage calls, affected supply nodes, expected restoration time, etc.

In an integrated environment, each system provides information to the other system so that they can operate together seamlessly.

- The outage system uses the set of current customers to determine and manage outages to minimize their impact.
- The outage system is informed of outages captured in Oracle Utilities Customer Care and Billing.
- Oracle Utilities Customer Care and Billing uses the current status of an outage at a given premise for customer service.

Customer Information Integration

The outage system needs information about current customers to determine and manage outages to minimize their impact. The current customer information in Oracle Utilities Customer Care and Billing must be made available in Oracle Utilities Network Management System. This can be done via data synchronization.

> **FASTPATH:**
> Refer to The Big Picture of Sync Requests for more information about synchronizing data.

Interfacing Outage Calls

The following points describe the integration:

- Oracle Utilities Customer Care and Billing is able to record trouble calls for a particular service point that exists in the system, as well as for an unknown service point, i.e. a fuzzy call. For a fuzzy call, the caller must provide either a street intersection, or a street segment.
- When an outage call is created and sent to the external system, an algorithm on the outage call business object is responsible for creating an outbound message that's sent to the external system. This is a real-time synchronous interface between Oracle Utilities Customer Care and Billing and Oracle Utilities Network Management System.
- Oracle Utilities Network Management System processes the Calls table and creates Incidents.

●

FASTPATH:
Refer to the *Oracle Utilities Customer Care and Billing - Network Management System Integration Implementation Guide* for information about outage call integration.

Outage Inquiry

Oracle Utilities Customer Care and Billing provides query transactions that can be used to make real-time synchronous calls to NMS and inquire on one of the following:

- Job History for a particular customer, service point, location or call identifier
- Call History for a particular customer, service point, location or call identifier
- Planned Outage Jobs for a particular service point

●

FASTPATH:
Refer to the *Oracle Utilities Customer Care and Billing - Network Management System Integration Implementation Guide* for information about outage query integration.

Setting Up The System To Enable Outage Integration

The following section provides an overview of how to enable the integration between Oracle Utilities Customer Care and Billing and Oracle Utilities Network Management System.

External System Setup

An external system must be setup in order integrate Oracle Utilities Network Management System. Once the external system is defined, specify this on the one NMS Integration feature configuration so the system knows which external system to use for outage queries.

Define Outbound Message Types

The following outbound message types are required for the integration:

- An outbound message type is required for each of the outage queries available
  - Job History
  - Call History
  - Planned Outages
- Once the outbound message types are defined, specify this on the one NMS Integration feature configuration so the system knows which outbound message types to use for outage queries.

- In addition, an outage call outbound message type is required for sending outage calls to Oracle Utilities Network Management System. This outbound message type must be referenced on your outage call types.

Define Characteristic Types

The following characteristic types must be defined to facilitate Outage integration:

Outage Group Code Characteristic Type

These characteristics are used to describe the outage problem.

- Create at least one pre-defined characteristic type
- For each characteristic type, define its list of valid values
- Include Service Task Type in the characteristic entity collection

> ➤ **NOTE:**
> **Characteristic Type Prefix**. The system attempts to build a dropdown list of your valid outage group codes when maintaining outage group types. To achieve this, all outage group code characteristic types must use the same prefix. This prefix must be defined on the NMS Integration feature configuration.
>
> **Integration with Outage Management**. The outage codes must be defined in both Oracle Utilities Customer Care and Billing and the outage management system. Refer to your Oracle Utilities Network Management System documentation for information about defining the outage codes there.

Contact Name Characteristic Type

This is used to link the contact name of the caller associated with an outage call as a characteristic on the outage call.

- Create an ad-hoc characteristic type
- Include Service Task Type in the characteristic entity collection
- Specify this characteristic type on the one NMS Integration feature configuration using the Outage Call Contact Name Characteristic Type feature option

Contact Number Characteristic Type

This is used to link the contact number of the caller associated with an outage call as a characteristic on the outage call.

- Create an ad-hoc characteristic type
- Include Service Task Type in the characteristic entity collection
- Specify this characteristic type on the one NMS Integration feature configuration using the Outage Call Contact Number Characteristic Type feature option

Call Identifier Characteristic Type

This is used to link the call identifier supplied by the caller associated with an outage call as a characteristic on the outage call.

- Create an ad-hoc characteristic type
- Include Service Task Type in the characteristic entity collection
- Specify this characteristic type on the one NMS Integration feature configuration using the Outage Call Identifier Characteristic Type feature option

Street Name Characteristic Type

This is used to link location information supplied by the caller associated with a fuzzy trouble call as a characteristic on the outage call.

- Create an ad-hoc characteristic type
- Include Service Task Type in the characteristic entity collection
- Specify this characteristic type on the one NMS Integration feature configuration using the Outage Call Street Name Characteristic Type feature option

Cross Street Name Characteristic Type

This is used to link location information supplied by the caller associated with a fuzzy trouble call as a characteristic on the outage call.

- Create an ad-hoc characteristic type
- Include Service Task Type in the characteristic entity collection
- Specify this characteristic type on the one NMS Integration feature configuration using the Outage Call Cross Street Name Characteristic Type feature option

Block Number Characteristic Type

This is used to link location information supplied by the caller associated with a fuzzy trouble call as a characteristic on the outage call.

- Create an ad-hoc characteristic type
- Include Service Task Type in the characteristic entity collection
- Specify this characteristic type on the one NMS Integration feature configuration using the Outage Call Block Number Characteristic Type feature option

City Characteristic Type

This is used to link location information supplied by the caller associated with a fuzzy trouble call as a characteristic on the outage call.

- Create an ad-hoc characteristic type
- Include Service Task Type in the characteristic entity collection
- Specify this characteristic type on the one NMS Integration feature configuration using the Outage Call City Characteristic Type feature option

State Characteristic Type

This is used to link location information supplied by the caller associated with a fuzzy trouble call as a characteristic on the outage call.

- Create an ad-hoc characteristic type
- Include Service Task Type in the characteristic entity collection
- Specify this characteristic type on the one NMS Integration feature configuration using the Outage Call State Characteristic Type feature option

NMS Integration - Feature Configuration

Create a feature configuration with the type NMS Integration. Populate entries for all the options.

NOTE:

Only one. The system expects only one NMS Integration feature configuration to be defined.

Configure the options for your interaction with the outage system.

➤ **NOTE:**
Your implementation may define additional options types. You do this by adding new lookup values to the lookup field NMS_OPT_TYP_FLG.

Option	Description
External System	This defines the external system used on outbound messages created when querying outage information in NMS from the outage management information portal page. Refer to Defining Service Order Options for more information. Only one value is allowed for this option.
Outbound Message Type - Call History	This defines the outbound message type used on outbound messages created when querying outage call history in NMS from the outage management information portal page. Refer to Defining Service Order Options for more information. Only one value is allowed for this option.
Outbound Message Type - Job History	This defines the outbound message type used on outbound messages created when querying outage job history in NMS from the outage management information portal page. Refer to Defining Service Order Options for more information. Only one value is allowed for this option.
Outbound Message Type - Planned Outages	This defines the outbound message type used on outbound messages created when querying planned outages in NMS from the outage management information portal page. Refer to Defining Service Order Options for more information. Only one value is allowed for this option.
Outage Group Code Characteristic Type Prefix	The system uses this prefix to populate the outage group code dropdown list during trouble call processing. Refer to Defining Service Order Options for more information. Only one value is allowed for this option.

Schema Constants - Feature Configuration

Create a feature configuration with the type Schema Constants. Populate entries for all the options listed below.

➤ **NOTE:**

Only one. The system expects only one Schema Constants feature configuration to be defined.

Configure the options for your interaction with the outage system.

➤ **NOTE:**
Your implementation may define additional options types. You do this by adding new lookup values to the lookup field F1CN_OPT_TYP_FLG.

Option	Description
Outage Call Contact Name Characteristic Type	This is used to link the contact name of the caller associated with an outage call as a characteristic on the outage call. Refer to Defining Service Order Options for more information. Only one value is allowed for this option.
Outage Call Contact Number Characteristic Type	This is used to link the contact number of the caller associated with an outage call as a characteristic on the outage call. Refer to Defining Service Order Options for more information. Only one value is allowed for this option.
Outage Call Identifier Characteristic Type	This is used to link the call identifier supplied by the caller associated with an outage call as a characteristic on the outage call. Refer to Defining Service Order Options for more information. Only one value is allowed for this option.
Outage Call Street Name Characteristic Type	This is used to link the street name supplied by the caller associated with a fuzzy trouble call as a characteristic on the outage call. Refer to Defining Service Order Options for more information. Only one value is allowed for this option.
Outage Call Cross Street Name Characteristic Type	This is used to link the cross street name supplied by the caller associated with a fuzzy trouble call as a characteristic on the outage call. Refer to Defining Service Order Options for more information. Only one value is allowed for this option.
Outage Call Block Number Characteristic Type	This is used to link the block number supplied by the caller associated with a fuzzy trouble call as a characteristic on the outage call. Refer to Defining Service Order Options for more information. Only one value is allowed for this option.
Outage Call City Characteristic Type	This is used to link the city supplied by the caller associated with a fuzzy trouble call as a characteristic on the outage call. Refer to Defining Service Order Options for more information. Only one value is allowed for this option.
Outage Call State Characteristic Type	This is used to link the state supplied by the caller associated with a fuzzy trouble call as a characteristic on the outage call. Refer to Defining Service Order Options for more information. Only one value is allowed for this option.

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

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
- **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:

- 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

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 **SPLService.**
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.

The Conversion Tool

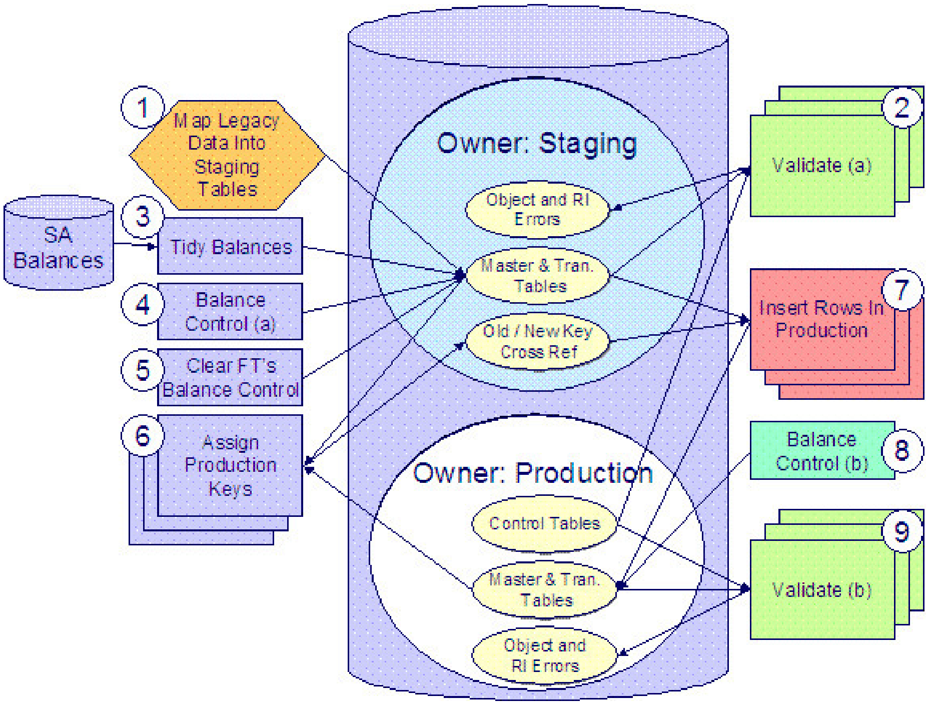
This section describes the Oracle Utilities Customer Care and Billing conversion tool.

Introduction

When you're ready to convert data from your legacy system into Oracle Utilities Customer Care and Billing, you will have analyzed your CIS processing requirements according to your business and organizational needs and set up the control tables accordingly.

FASTPATH:
Refer to the **Administration Guide** for a complete discussion of the various control tables and the order in which they must be set up.

After the control tables are set up, you are ready to load data into the system from your legacy system. This conversion effort involves several steps as illustrated in the following diagram:



The following points briefly outline each of the above tasks:

- **Map Legacy Data Into Staging.** During this step, your legacy master data (e.g., account, person, premise, meter) and transaction data (e.g., bills, payments, meter reads) is migrated 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. Refer to Appendix B - Multiple Owners In A Single Database for information about table ownership.

WARNING:
The above diagram illustrates how the system is configured to support the conversion effort in the standard installation, i.e., the staging tables are in the same database as the production tables (each with a different owner). However, it is possible for the staging tables to be in a separate database. This option requires additional effort on your part (because you would have to copy the control tables from production into your staging database). Please refer to Appendix B - Multiple Owners In A Single Database for information about this alternative.

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 (a).** During the validation (a) step, the system validates the data you loaded into the staging tables. Two types of validation programs exist:
  - **Object Validation Programs.** Each of the system's master data objects (e.g., person, account, premise, meter, etc.) is validated using the same logic that is used to validate data added by users in your production system.
  - **Referential Integrity Validation Programs.** After you have successfully validated the master data objects, the referential integrity validation programs are executed to validate transaction data and to highlight "orphaned" rows. These programs check the validity of the foreign keys on all rows on all tables.

➤ **NOTE:**
Control tables from production. It's important to notice that the validation programs validate your staging data using the control tables that have been set up in production. Refer to Appendix B - Multiple Owners In A Single Database for a description of how this works.

- **Tidy Balances.** During this step, the system creates adjustments that cause each SA's current and payoff balances to equal the desired balances. The desired balances are supplied on a flat file prepared by you.
- **Balance Control (a).** During this step, you run the balance control program and then verify that the balances that it generates are consistent with the balances in your legacy system.
- **Clear FT's Balance Control.** In the previous step, the system creates a balance control and links it to the FT's. If the balance control's balances are consistent with the amount of receivables being transferred into the system, you should run the Clear FT's Balance Control program. This program simply resets the Balance Control column on the FT so that the FT's can be included in a balance control (see the last step below) after they have been transferred to production.
- **Assign Production Keys.** During this step, the system allocates random, clustered keys to the rows in the staging database.
- **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 are reassigned using the data populated in the previous step.
- **Balance Control (b).** During this step, you run the balance control program against production. You do this to verify the balances in production are consistent with the values of receivables converted from your legacy application.
- **Validate (b).** During this step, you rerun the object validation programs, but this time against production. We recommend rerunning these programs to confirm that the insertion programs have executed successfully. We recommend running these programs 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 programs in full validation mode (to validate every object).

Conversion Tool Steps

The following sections provide more details about the steps in the conversion process.

Map Legacy Data Into Staging Tables

This section provides some high level discussion about mapping legacy data to the system's staging tables. Refer to The Staging Tables for details about the staging tables in the system.

> **NOTE:**
> **Recommendation.** You can use any method you prefer to load Oracle Utilities Customer Care and Billing 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.

Populating Characteristic Tables. There are many maintenance objects that include a characteristic table used to capture miscellaneous information about the object, e.g. Person, Account, Service Agreement, etc. 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.

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 account ID is allocated, the key assignment program checks the account key table to see if it exists.
- 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 CI_PREM is CI_PREM_K
- The name of every table's key table is defined under the Generated Keys column in the Table Names sections in The Staging Tables.
- 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.

Validate Information In The Staging Tables

During the first validation step, the system validates the data you loaded into the staging tables. Two types of validation programs exist:

- **Object Validation Programs.** The object validation programs validate each of the system's master data objects (e.g., person, account, premise, meter, etc.) and a limited number of transaction data objects (e.g., field activity, field order, etc.). Please note that these programs call the same programs that are used to validate data added by users in your production system.
- **Referential Integrity Validation Programs.** After the master data objects have been validated, the referential integrity validation programs are executed to validate transaction data and to highlight "orphaned" rows. These programs simply check the validity of the foreign keys on all rows on all tables.

The contents of this section describe how to execute the validation programs.

Object Validation Programs

Each of the objects described under Master Data must be validated using the respective object validation program indicated in its Table Names section.

In a limited number of cases object validation is available for Transaction Data objects, where customers may convert transaction data that is still pending. For example, if you are converting pending field activities, you want to ensure that the data is valid. For these cases you may also be converting historic records. For example, in addition to the pending field activities you are converting completed field activities to keep a historic view. You may not want to perform validation on completed records. As a result the background processes provided for transaction data allow you to limit the validation to records in a give status.

We strongly recommend validating each object in the following steps:

- Execute each object's validation program in random-sample mode to highlight pervasive errors. When you execute a validation in random-sample mode, you are actually telling it to validate every X records (where X is a parameter that you supply to the job). Refer to Submitting Object Validation Programs for more information about the parameters supplied to these background processes.
- You can view errors highlighted by validation programs using the Validation Error Summary transaction.

- Correct the errors using SQL. Note, you can use the base package's transactions (e.g., Person Maintenance, Premise Maintenance, etc.) 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 program in all-instances mode to highlight elusive, one-off errors. Refer to Submitting Object Validation Programs for more information about the parameters supplied to these background processes.

WARNING:
Whenever an object validation program is run, it is necessary to delete all previously recorded errors associated with its tables from the validation error table before it inserts new errors.

After the various object validation programs run cleanly, run the referential integrity validation programs as described in the next section.

➤ **NOTE:**
Another use for these programs. In addition to validating your objects after conversion or an upgrade, the validation programs are another use. Say for example, you want to experiment with changing the validation of a person and you want to determine the impact of this new validation on your existing persons. You could change the validation and then run the person validation object - it will produce errors for each person that fails the new validation.

Submitting Object Validation Programs

The object validation programs that are described in the staging tables table names matrices are classic background processes as they can also be run against production data. You submit these processes in the same way you submit any background process in production.

Referential Integrity Validation Programs

It's important to understand that only master data objects (e.g., persons, accounts, meters, premises, etc.) are validated by the object validation programs discussed above. This means that only master data objects will have their foreign keys checked for validity by the object validation programs. You must run the referential integrity programs to validate all other data.

The referential integrity validation programs highlight:

- Orphaned rows because orphan rows, by definition, don't reference an object.
- Invalid foreign keys on transaction data.

➤ **NOTE:**
Validating Transaction Data. You may wonder why transaction data is not subject to the object validation routines. This is because: a) the production system only needs validation logic for master data because transaction data is not entered by users, and b) most conversions necessitate loading skeletal transaction data because the legacy system typically doesn't contain enough information to create accurate transactions in the system.

Each of the tables described under Transaction Data must be validated using the respective referential integrity validation program indicated in its Table Names section. We strongly recommend validating each table in the following steps:

- Execute each table's referential integrity validation program. Refer to Submitting Referential Integrity Validation Programs for more information about the parameters supplied to these background processes.
- You can view errors highlighted by this process using the FK Validation Summary transaction.
- Correct the errors using SQL (you cannot use the application to correct these types of errors).
- Rerun the referential integrity programs until no errors are produced.

WARNING:
Whenever you run a referential integrity validation program, it deletes all errors associated with its table from the referential integrity error table.

In order to highlight orphaned rows in the master data, run the referential integrity validation programs against all tables described under Master Data using the procedure described above.

When ALL referential integrity programs indicate the staging database is clean, you may now proceed to the next step - tidy balances.

Submitting Referential Integrity Validation Programs

The referential integrity validation programs described under Master Data and Transaction Data (in the Table Names matrices) are submitted using a batch driver program, CIPVRNVB, and this program is executed in the staging database. Please note that the referential integrity validation programs may also be run in the production environment on occasion, to determine the integrity of data in the production database.

You should supply the following parameters to this program:

- **Batch code.** The batch code associated with the appropriate table's referential integrity validation program. Refer to each table listed under Master Data and Transaction Data (in the Table Names matrices) for each referential integrity batch code / program.
- **Batch thread number.** Thread number is not used and should be left blank.
- **Batch thread count.** Thread count is not used and should be left blank.
- **Batch rerun number.** Rerun number is not used and should be left blank.
- **Batch business date.** Business date is the date supplied to the referential integrity validation programs and the date under which statistics will be logged.
- **Total number of commits.** Total number of commits is not used and should be left blank.
- **Maximum minutes between cursor re-initiation**. Maximum minutes between cursor re-initiation is not used and should be left blank.
- **User ID.** User ID is only used to log statistics for the execution of the batch job.
- **Password.** Password is not 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 at start** (Y/N), **trace program exit** (Y/N), **trace SQL** (Y/N) and **output trace** (Y/N). These switches are only used during QA and benchmarking. If trace program start is set to Y, a message is displayed whenever a program is started. If trace program at exist is set to Y, a message is displayed whenever a program is exited. If trace SQL is set to Y, a message is displayed whenever an SQL statement is executed.

Recommendations To Speed Up Validation Programs

The following points describe ways to accelerate the execution of the validation programs:

- Ensure that statistics are recalculated after data has been inserted into the staging tables. For Oracle users, we strongly recommend using the Oracle-provided PL/SQL package to generate statistics rather than the analyze command.
- Object validation programs should be run 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.
- Referential integrity validation programs run fairly quickly without much tuning. However, additional benefits are gained by running several programs at the same time.
- Remember that the object validation programs can be run in "validate every n th row". We recommend running these programs using a largish value for this parameter until the pervasive problems have been rectified.

Tidy Balances

This background process creates adjustments that cause each SA's current and payoff balances to equal its balance in the legacy system (note: the batch control ID of CNV-BAL is used for this process).

NOTE:
Submitting this process. You submit this process in the staging database. Refer to Tidy Balances for a description of this process and its parameters.

You supply the desired balances to this background process in a flat file in the following format:

Field	Size	Description
SA ID	X10	The unique identifier of the service agreement
Payoff Balance Sign	X1	Positive or Negative value indicates debit or credit balance respectively
Payoff Balance	N15.2	The SA's payoff balance (how much the customer really owes).
Current Balance New Charge Sign	X1	Positive or Negative value indicates debit or credit balance respectively
Current Balance - New Charge	N15.2	The amount of the SA's current balance that is considered a new charge, i.e., it hasn't started aging yet.
Current Balance Amount 1 Sign	X1	Positive or Negative value indicates debit or credit balance respectively
Current Balance - Amount 1	N15.2	The amount of the SA's current balance that is X days old (X is defined in the next field)
Age of Current Balance - Amount 1	N3	The number of days old the prior field is (if you keep your debt in "buckets" as opposed

		to knowing the exact number of days it has aged, you will have to choose an exact age). Set this value to zero if the value of amount 1 should be considered a "new charge" (i.e., it should only start aging when it is swept onto the next bill produced for the SA's account)
Current Balance Amount 2 Sign	X1	Positive or Negative value indicates debit or credit balance respectively
Current Balance - Amount 2	N15.2	The amount of the SA's current balance that is X days old (X is defined in the next field)
Age of Current Balance - Amount 2	N3	The number of days old the prior field is (if you keep your debt in "buckets" as opposed to knowing the exact number of days it has aged, you will have to choose an exact age)
Current Balance Amount 3 Sign	X1	Positive or Negative value indicates debit or credit balance respectively
Current Balance - Amount 3	N15.2	The amount of the SA's current balance that is X days old (X is defined in the next field)
Age of Current Balance - Amount 3	N3	The number of days old the prior field is (if you keep your debt in "buckets" as opposed to knowing the exact number of days it has aged, you will have to choose an exact age)
Current Balance Amount 4 Sign	X1	Positive or Negative value indicates debit or credit balance respectively
Current Balance - Amount 4	N15.2	The amount of the SA's current balance that is X days old (X is defined in the next field)
Age of Current Balance - Amount 4	N3	The number of days old the prior field is (if you keep your debt in "buckets" as opposed to knowing the exact number of days it has aged, you will have to choose an exact age)
Current Balance Amount 5 Sign	X1	Positive or Negative value indicates debit or credit balance respectively
Current Balance - Amount 5	N15.2	The amount of the SA's current balance that is X days old (X is defined in the next field)
Age of Current Balance - Amount 5	N3	The number of days old the prior field is (if you keep your debt in "buckets" as opposed to knowing the exact number of days it has aged, you will have to choose an exact age)

NOTE:

Submitting this process. You submit this process in the staging database. Refer to Tidy Balances for a description of this process and its parameters.

Balance Control (a)

During this step, you run the balance control programs and then verify that the balances that it generates are consistent with the balances in your legacy system.

➤ **NOTE:**
Submitting this process. You submit this process in the staging database. Refer to The Big Picture of Balance Control for more information about the balance control processes. Refer to Balance Control for information about the page used to view the balances generated by this process.

Clear FT Balance Control

In the previous step, the system created a balance control and links it to the FT's. If the balance control's balances are consistent with the amount of receivables being transferred into the system, you should run the Clear FT's Balance Control program. This program simply resets the Balance Control column on the FT so that the FT's can be included in a balance control (see the last step below) after they have been transferred to production. Note: the batch control ID of CNV-BCG is used to request this process.

➤ **NOTE:**
Submitting this process. You submit this process in the staging database. Refer to Reset Balances for a description of this process and its parameters.

Allocate Production Keys

The topics in this section describe the background processes used to assign production keys to the staging data.

The Big Picture of Key Assignment

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 old key / new key tables are named the same as their primary table, but rather than being prefixed by "CI", they are prefixed by "CK". For example, the old key / new key table for CI_ACCT is called CK_ACCT.

The insertion programs that transfer 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. Note that the capability of assigning the new key to a foreign key applies to

- "True" foreign keys, i.e. where the key is a column in another table. For example, CI_SA has a column for ACCT_ID.
- FK reference characteristics. These are characteristics that define, through an FK reference, the table and the key that this characteristic represents.

The insertion programs are not able to assign "new keys" to foreign keys defined in an XML structure field (CLOB).

The key assignment programs listed under Master Data and Transaction Data (in the table names sections) are responsible for populating the old key / new key tables (i.e., you don't have to populate these tables). Because the population of the rows in these tables is IO intensive, we have supplied detailed instructions that will accelerate the execution time of these programs.

> **NOTE:**
> **Why are keys reassigned?** 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).
>
> **Iterative conversions.** Rather than perform a "big bang" conversion (one where all customers are populated at once), some implementations have the opportunity to go live on subsets of their customer 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 keys.
>
> **Program Dependencies**. The programs used to assign production keys are listed in the Table Names matrices. Most of these programs have no dependencies (i.e., they can be executed in any order you please). The exceptions to this statement are noted in Program Dependencies.

Submitting Key Assignment Programs

The key assignment programs described under Master Data and Transaction Data (in the Table Names matrices) are submitted using a batch driver program, CIPVRNKB, and this program is executed in the staging database. You should supply the following parameters to this program:

- **Batch code.** The batch code associated with the appropriate table's key assignment program. Refer to each table listed under Master Data and Transaction Data (in the Table Names matrices) for each key assignment batch code / program.
- **Batch thread number.** Thread number is not used and should be left blank.
- **Batch thread count.** Thread count is not used and should be left blank.
- **Batch rerun number.** Rerun number is not used and should be left blank.
- **Batch business date.** Business date is the date supplied to the key assignment programs and the date under which statistics will be logged.
- **Total number of commits.** Total number of commits is not used and should be left blank.
- **Maximum minutes between cursor re-initiation**. Maximum minutes between cursor re-initiation is not used and should be left blank.
- **User ID.** User ID is only used to log statistics for the execution of the batch job.
- **Password.** Password is not 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 at start** (Y/N), **trace program exit** (Y/N), **trace SQL** (Y/N) and **output trace** (Y/N). These switches are only used during QA and benchmarking. If trace program start is set to Y, a message is displayed whenever a program is started. If trace program at exist is set to Y, a message is displayed whenever a program is exited. If trace SQL is set to Y, a message is displayed whenever an SQL statement is executed.
- **Mode.** The proper use of this parameter will greatly speed up the key assignment step as described under Recommendations To Speed Up Key Generation. This parameter has three values:

- If you supply a mode with a value of I (initial key generation), the system allocates new keys to the rows in the staging tables (i.e., it populate the respective old key / new key table).
- If you supply a mode with a value of D (resolve duplicate keys), the system reassigns keys that are duplicates.
- If you supply a mode with a value of B (both generate keys and resolve duplicates), the system performs both of the above steps. This is the default value if this parameter is not supplied.
- Please see Recommendations To Speed Up Key Generation for how to use this parameter to speed up the execution of these processes.

> **NOTE:**
> **Parallel Key Generation.** No key generation program should be run (either in mode I or B) while another program is being run unless that program is in the same tier (see Program Dependencies for a description of the tiers).

• **Start Row Number.** This parameter is only used if you are performing conversions where data already exists in the tables in the production database (subsequent conversions). In an Oracle database the key assignment routines create the initial values of keys by manipulation of the Oracle 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 purpose of this parameter is to increase the value of row number by the given value, and minimize the chance of duplicate key assignment.

Recommendations To Speed Up Key Generation Programs

The following points describe ways to accelerate the execution of the key generation programs.

> **NOTE:**
> **Naming convention.** The convention "CK_<table_name>" is used to denote the old key / new key tables described under The Big Picture of Key Assignment.

- 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 CK_<table_name> 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 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. For Oracle users, we strongly recommend using the Oracle-provided PL/SQL package to generate statistics rather than the analyze command.

- Optimal use of the **Mode** parameter under Submitting Key Assignment Programs.
  - Before any key assignments, alter both the "old key" CX_ID index and the "new key" CI_ID index on the CK_<table_name> tables to be unusable.
  - Run all key assignment tiers, submitting each job with MODE = "I".
  - Rebuild the CX_ID and CI_ID indexes on the CK_<table_name>. Rebuilding the indexes using both the PARALLEL and NOLOGGING parameters will speed the index creation process in an Oracle DB. Statistics should be computed for these indexes.
  - Run all key assignment tiers that were previously run in MODE = 'I', submitting each job with MODE = "D". This will reassign all duplicate keys.

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.

The Big Picture Of Insertion Programs

All insertion programs are independent and may run concurrently. Also note, all insertion programs can be run in many parallel threads as described in the next section (in order to speed execution).

Submitting Insertion Programs

The insertion programs described under Master Data and Transaction Data (in the Table Names matrices) are submitted using a batch driver program, CIPVRNIB, and this program is executed in the staging database. You should supply the following parameters to this program:

- **Batch code.** The batch code associated with the appropriate table's insertion program. Refer to each table listed under Master Data and Transaction Data (in the Table Names matrices) for each insertion batch code / program.
- **Batch thread number.** Thread number contains the relative thread number of the process. For example, if you want to insert accounts into production in 20 parallel threads, each of the 20 execution instances receives its relative thread number (1 through 20). Refer to Parallel Background Processes for more information.
- **Batch thread count.** Thread count contains the total number of parallel threads that have been scheduled. For example, if the account insertion process has been set up to run in 20 parallel threads, each of the 20 execution instances receives a thread count of 20. Refer to Parallel Background Processes for more information.
- **Batch rerun number.** Rerun number is not used and should be left blank.
- **Batch business date.** Business date is the date supplied to the insertion programs and the date under which statistics will be logged.
- **Total number of commits.** This is the number of commits IN TOTAL that you want to perform. For example, if you have 1,000,000 accounts and you supply a value of 100; then a commit will be executed for approximately every 10,000 accounts.
- **Maximum minutes between cursor re-initiation**. This should only be populated if you want to override the default value of 15.

- **User ID.** User ID is only used to log statistics for the execution of the batch job.
- **Password.** Password is not 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 at start** (Y/N), **trace program exit** (Y/N), **trace SQL** (Y/N) and **output trace** (Y/N). These switches are only used during QA and benchmarking. If trace program start is set to Y, a message is displayed whenever a program is started. If trace program at exist is set to Y, a message is displayed whenever a program is exited. If trace SQL is set to Y, a message is displayed whenever an SQL statement is executed.

Recommendations To Speed Up Insertion Programs

The following points describe ways to accelerate the execution of the insertion programs:

- Before running the first insertion program:
  - Rebuild the index on the prime key on the old key / new key table (i.e., those tables prefixed with "CK").
  - Re-analyze the statistics on the old key / new key table (i.e., those tables prefixed with "CK"). For Oracle users, we strongly recommend using the Oracle-provided PL/SQL package to generate statistics rather than the analyze command.
  - 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. For Oracle users, we strongly recommend using the Oracle-provided PL/SQL package to generate statistics rather than the analyze command.

Run Balance Control Against Production

During this step, you rerun the balance control program, but this time against production. You do this to verify the balances in production are consistent with the values of receivables converted from your legacy application.

> **NOTE:**
> **Submitting this process.** You submit this process in the production database. Refer to The Big Picture of Balance Control for more information about the balance control processes. Refer to Balance Control for information about the page used to view the balances generated by this process.

Validate Production

During this step, you rerun the object validation programs, but this time in production. We recommend rerunning these programs to confirm that the insertion programs have executed successfully. We recommend running these programs in random sample mode (e.g., validate every 1000 th object) rather than conducting a full validation in order to save time. However, if you have time, you should run these programs in full validation mode (to validate every object). Please refer to the various "Table Names" sections above for the respective names of the programs to run.

The Validation User Interface

The topics in this section describe the various pages that assist in the conversion effort.

Validation Error Summary

Navigate to **Admin** > **Conversion** > **Validation Error Summary** to view validation errors associated with the objects defined in Master Data.

Description of Page

You can use **Table Name** to restrict errors to a specific object. If this field is left blank, all errors on all objects will be displayed.

The grid contains a separate row for each type of error. The following information is displayed:

- **Table Name** is the name of the main table associated with the object.
- **Message Category** and **Message Number** define the type of error. These fields are the unique identifier of the message that describes the error (the verbiage of this message is displayed in the **Message Text** column).
- **Count** contains the number of records with this error. Press the Go To button to see the individual records with the error.

Validation Error Detail

This page is used to view validation errors of a given type associated with one of the objects defined in Master Data. This transaction is not intended to be invoked from the **Admin** menu. Rather, drill into the validation details from Validation Error Summary.

Description of Page

Use **Table Name**, **Message Category**, and **Message Number** to define the object and the type of error you wish to display. The grid contains a separate row for each object with the given type of error. The following information is displayed:

- **Table Name** is the name of the main table associated with the object.
- **Record Identifier** is the unique identifier of the object with the error (e.g., the person ID, the account ID, the premise ID, etc.). Press the Go To button to transfer to the maintenance page associated with the object.
- **Message Category** and **Message Number** define the type of error. These fields are the unique identifier of the message that describes the error (the verbiage of this message is displayed in the **Message Text** column).

FK Validation Summary

Navigate to **Admin** > **Conversion** > **FK Validation Summary** to view foreign key validation errors associated with the objects defined in Master Data.

Description of Page

You can use **Table Name** to restrict errors to a specific object. If this field is left blank, all errors on all objects will be displayed.

The grid contains a separate row for each type of error. The following information is displayed:

- **Table Name** is the name of the main table associated with the object.
- **Count** contains the number of records on this table that have this error.
- **Foreign Key Field Names 1 to 6** contain the names of the foreign keys contained on this table that have been found to be in error.
- **Foreign Key Values 1 to 6** contain the values within the foreign key fields that are found to be in error.

FK Validation Detail

This page is used to view foreign key validation errors of a given type associated with one of the objects defined in Master Data. This transaction is not intended to be invoked from the **Admin** menu. Rather, drill into the validation details from FK Validation Summary.

Description of Page

Use **Table Name** to specify the table you wish to view. The names and values of the foreign key fields on the table are displayed. The grid that follows contains the primary key values of this table's records that are in error. The following information is displayed:

- **Table Name** is the name of the main table.
- **FK Fields 1 to 6** are the names of the foreign keys contained in this table. Displayed alongside the key names are the values within these fields. These identify records on other tables to which this tables record is related. For example, the CI_PREM_GEO record identified by its displayed primary keys should be related to a Premise record with the Premise ID shown - it appears in this list only if there is something amiss with this relationship.

The Staging Tables

This section describes the objects into which your legacy data is mapped. For each object, we provide the following:

- A data model.
- An indication of which tables have system-assigned keys.
- The physical table names.
- The name of the batch control to submit to validate the object.
- The name of the program (and related batch control) that validates each table for referential integrity.
- The name of the program (and related batch control) that performs key assignment for each table.
- The name of the program (and related batch control) that inserts the table's rows into production from staging.
- Suggestions to assist in the conversion process.

WARNING:
We recommend you read this document on a browser (or using Word under windows) so you can take advantage of the Color Coding.

NOTE:
Column details do not appear in this document. When you're ready to examine an object's tables, use the hyperlinks in the respective Table Names section to transfer to the data dictionary. The data dictionary will

show you the required columns, the foreign keys (and their related tables), the source code of the program that validates the contents of the table, and a host of other information that will assist the conversion process.

WARNING:
In the data models that appear below, you will find a variety of entities that are classified as either a control table or a lookup table. Please refer to Color Coding for more information about how to recognize such an entity.

A Note About Programs in the Table Names Matrices

For each object described in the master data and transaction data sections, there is a "table names" section that includes a matrix listing the name of each table that is part of the maintenance object. Included in the matrix is information about the programs provided to perform object validation, referential integrity validation, key assignment and insertion. The following are some points about these programs:

- One object validation program exists for the entire set of tables for the maintenance object. The **Object Validation Batch Control** column indicates the batch control used to submit the object validation. Refer to Submitting Object Validation Programs for more information. Drilling down on the hypertext allows you to see more information about the batch control, including the program associated with it.
- A referential integrity validation program exists for every table whose key includes a parent key of another object. As described in Submitting Referential Integrity Validation Programs, these programs are submitted using a driver supplied by the system where the batch code for the appropriate table is provided. (The driver then executes the program whose name matches the batch code). The **Referential Integrity Validation Batch Control** column indicates the table's batch control / program name.
- One key assignment program exists for the parent table for the maintenance object. As described in Submitting Key Assignment Programs, these programs are submitted using a driver supplied by the system where the batch code for the appropriate table is provided. (The driver then executes the program whose name matches the batch code). The **Key Assignment Batch Control** column indicates the table's batch control / program name.
- An insertion program exists for every table for the maintenance object. As described in Submitting Insertion Programs, these programs are submitted using a driver supplied by the system where the batch code for the appropriate table is provided. (The driver then executes the program whose name matches the batch code). The **Insertion Batch Control** column indicates the table's batch control / program name.

Master Data

This section describes the various "master data" objects (e.g., person, account, meter, etc.) that must be created before you can convert transaction data.

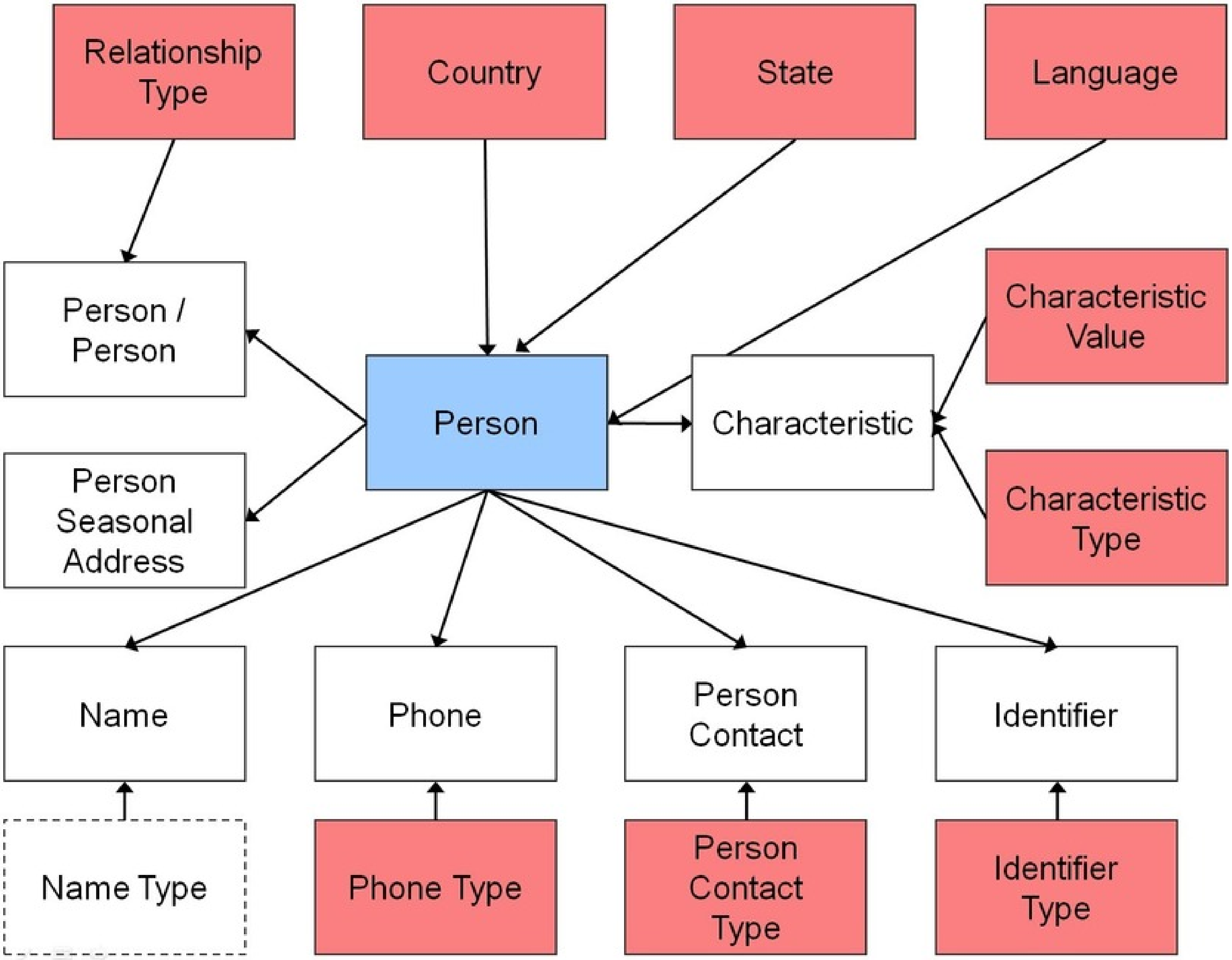
➤ **NOTE:**
Key Assignment Dictates The Order Of Conversion. The following contents are listed in the order in which the objects should be converted in order to maintain referential integrity.

Person

Each customer must have a person and an account object. This section describes the person object. Refer to Account for details about the account object.

Person Data Model

The following data model illustrates the person object.



Person Table Names

Data Model Name	Table Name	Generated Keys	Object Validation Batch Control	Referential Integrity	Key Assignment Batch Control	Insertion Batch Control

				Validation Batch Control		
Person	CI_PER	Yes CI_PER_K	VAL-PER		CIPVPERK	CIPVPERI
Name	CI_PER_NAME	No. The key is PER_ID plus a sequence number.		CIPVPNMV		CIPVPNMI
Person / Person	CI_PER_PER	No. The key is PER_ID1, PER_ID2, relationship type and start date.		CIPVPPEV		CIPVPPEI
Phone	CI_PER_PHONE	No. The key is PER_ID plus phone type.		CIPVPPHV		CIPVPPHI
Identifier	CI_PER_ID	No. The key is PER_ID plus identifier type.		CIPVPIDV		CIPVPIDI
Characteristic	CI_PER_CHAR	No. The key is PER_ID plus an edate and a char type.		CIPVPRCV		CIPVPRCI
Seasonal Address	CI_PER_ADDR_SEAS	No. The key is PER_ID plus a sequence number.		CIPVPSAV		CIPVPSAI

Person Suggestions

A person must have at least one row on the name table and at least one of the names must be marked as being the primary name.

A person must have at least one row on the identity table and at least one of the identities must be marked as being the primary ID.

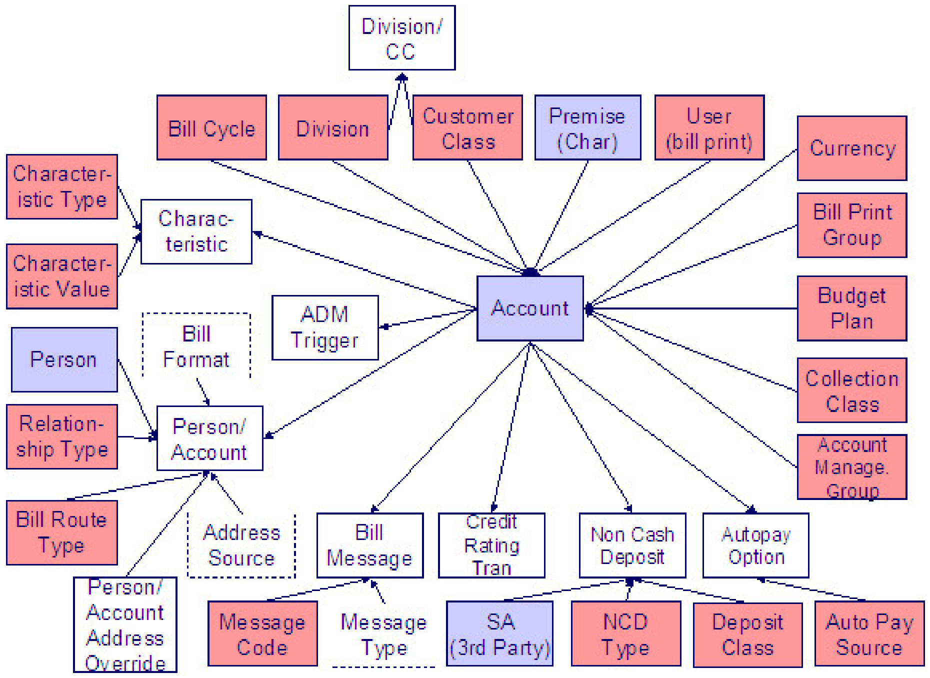
The country and state are only necessary if the person has an override mailing address.

Account

Each customer must have a person and an account object. This section describes the account object, refer to Person for details about the person object.

Account Data Model

The following data model illustrates the account object.



Account Table Names

Data Model Name	Table Name	Generated Keys	Object Validation Batch Control	Referential Integrity Validation Batch Control	Key Assignment Batch Control	Insertion Batch Control
Account	CI_ACCT	Yes CI_ACCT_K	VAL-ACCT		CIPVACCK	CIPVACCI
Bill Message	CI_ACCT_MSG	No. The key is account ID plus bill message code.		CIPVMSGV		CIPVMSGI
Autopay Option	CI_ACCT_APAY	Yes CI_ACCT_APAY_K		CIPVAAPV	CIPVAAPK Has dependencies	CIPVAAPI
Characteristic	CI_ACCT_CHAR	No. The key is ACCT_ID plus an edate and a char type.		CIPVACHV		CIPVACHI
Person/Account	CI_ACCT_PER	No. The key is account ID plus person ID.		CIPVACPV		CIPVACPI
Person/Account Address Override	CI_PER_ADDR_OVRD	No. The key is Account ID plus Person ID		CIPVPAOV		CIPVPAOI
Non Cash Deposit	CI_NCD	No. The key is account ID plus seq number		CIPVNCDV		CIPVNCDI

Credit Rating Tran	CI_CR_RAT_HIST	Yes CI_CR_RAT_HIST_K	CIPVCRTV	CIPVCRRK Has dependencies	CIPVCRTI
ADM Trigger	CI_ADM_RVW_SCH	No. The key is account ID plus date	CIPVARSV		CIPVARSI

Account Suggestions

An account must have at least one row on the account / person table and at least one account / person must be marked as being the main customer. Please see column notes for the account / person table for inter-field validation in respect of the various switches (e.g., if main customer switch is on, then the person must also be financially responsible).

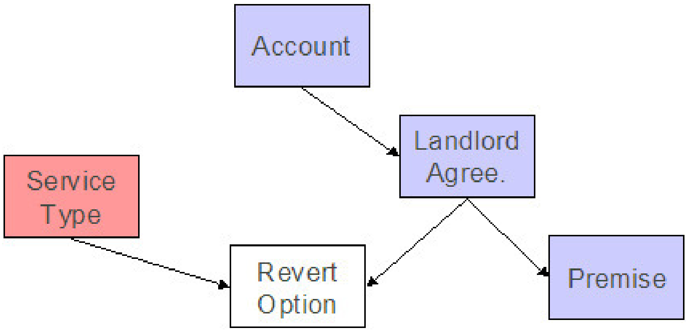
We recommend storing an ADM trigger (CI_ADM_RVW_SCH) for every account where the trigger date is the conversion date. This will cause the account to be reviewed by the account debt monitor when it next runs. We have supplied a dedicated batch process for this purpose that simply inserts a row in this table with the review date set equal to the current date. This will ensure that all converted accounts are reviewed after they are inserted into production. This program is named CIPVADMB and goes by the batch control ID of CNV-ADM.

If your legacy system has the equivalent of a credit rating or a cash only score, you should create credit rating transactions. The values you create need to be consistent with the beginning and threshold credit rating and cash-only points on the installation record or the account’s CIS Division. Refer to the business process guide - customer information - how are credit rating transactions created for more information.

Landlord

Landlord Data Model

The following data model illustrates the landlord object.



Landlord Table Names

Data Model Name	Table Name	Generated Keys	Object Validation Batch Control	Referential Integrity Validation Batch Control	Key Assignment Batch Control	Insertion Batch Control
Landlord	CI_LANDLORD	Yes CI_ LANDLORD_K	VAL-LL		CIPVLNDK	CIPVLNDI
Revert Option	CI_LL_DETAIL	No. The key is LL_ID plus service type		CIPVLLDV		CIPVLLDI

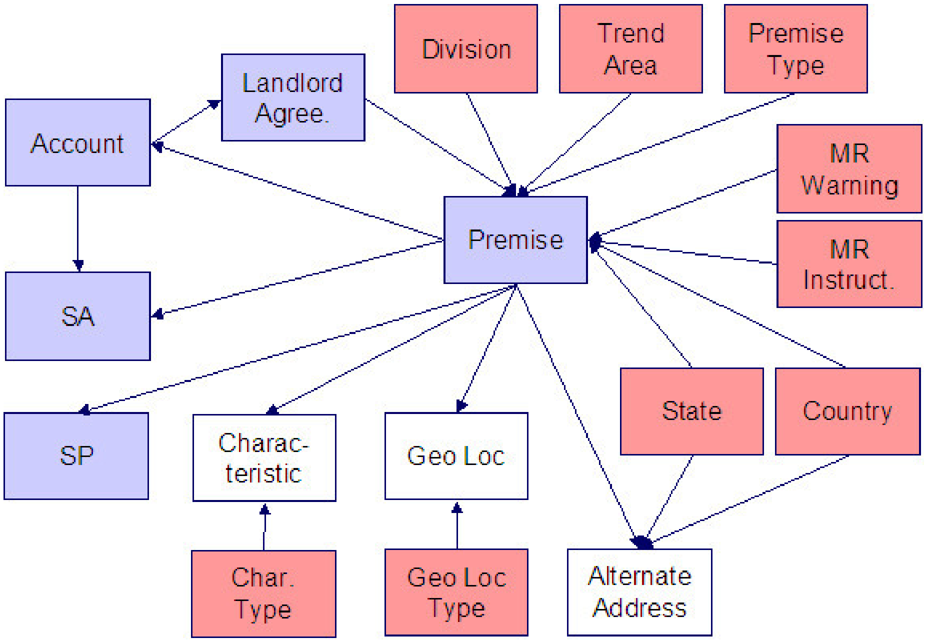
Landlord Suggestions

N/A.

Premise

Premise Data Model

The following data model illustrates the premise object.



NOTE: Trend Area is not applicable for Oracle Utilities Customer to Meter.

Premise Table Names

Data Model Name	Table Name	Generated Keys	Object Validation Batch Control	Referential Integrity Validation Batch Control	Key Assignment Batch Control	Insertion Batch Control
Premise	CI_PREM	Yes CI_PREM_K	VAL-PREM		CIPVPRMK Has dependencies	CIPVPRMI
Characteristic	CI_PREM_CHAR	No. The key is PREM_ID plus an edate and a char type.		CIPVPCHV		CIPVPCHI
Geo Loc	CI_PREM_GEO	No. The key is PREM_ID plus geo loc type.		CIPVPGOV		CIPVPGOI
Alternate Address	CI_PRM_ALT_ADDR	Yes CI_PRM_ALT_ADDR_K		CIPVAPAV	CIPVAPAK Has dependencies	CIPVAPAI

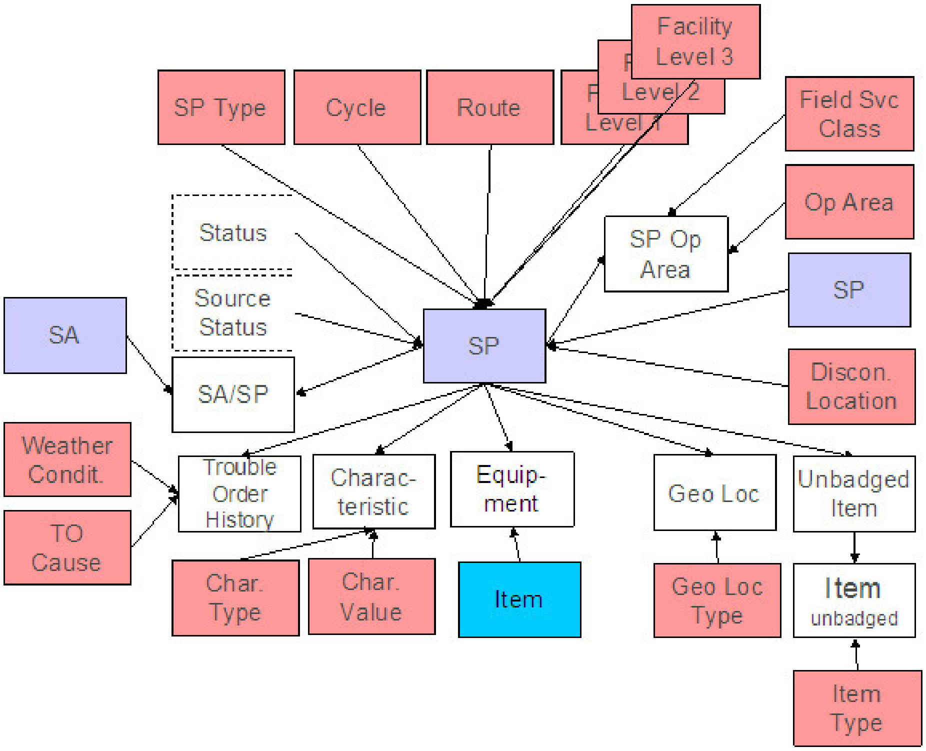
Premise Suggestions

N/A.

Service Point

Service Point Data Model

The following data model illustrates the service point object.



➤ **NOTE:** Equipment, Facility Level,, Operation Area, and Service Class are not applicable for Oracle Utilities Customer to Meter

Service Point Table Names

Data Model Name	Table Name	Generated Keys	Object Validation Batch Control	Referential Integrity Validation Batch Control	Key Assignment Batch Control	Insertion Batch Control
Service Point	CI_SP	Yes CI_SP_K	VAL-SP		CIPVSPPK Has dependencies	CIPVSPPI
Characteristic	CI_SP_CHAR	No. The key is SP_ID plus an edate and a char type.		CIPVSPCV		CIPVSPCI
Equipment (not applicable to Oracle Utilities Customer Care and Billing)	CI_SP_EQ	No. The key is service point ID plus equipment (service point) ID.		CIPVSEQV		CIPVSEQI
Geo Loc	CI_SP_GEO	No. The key is service point ID plus geo type.		CIPVSPGV		CIPVSPGI

Unbadged Item	CI_SP_MULT_ITEM	No. The key is SP_ID plus edate.		CIPVSPMV		CIPVSPMI
Item	CI_MULT_ITEM	No. The key is SP_ID, edate and item type.		CIPVSMIV		CIPVSMII
SP Op Area (not applicable to Oracle Utilities Customer to Meter)	CI_SP_OP_AREA	No. The key is SP_ID plus field service classification code		CIPVSPOV		CIPVSPOI
SA/SP	CI_SA_SP (note, this table really belongs to the SA object, it is included here for completeness)	Yes CI_SA_SP_K	CIPVSVAB	CIPVSAPV	CIPVSSPK Has dependencies	CIPVSAPI

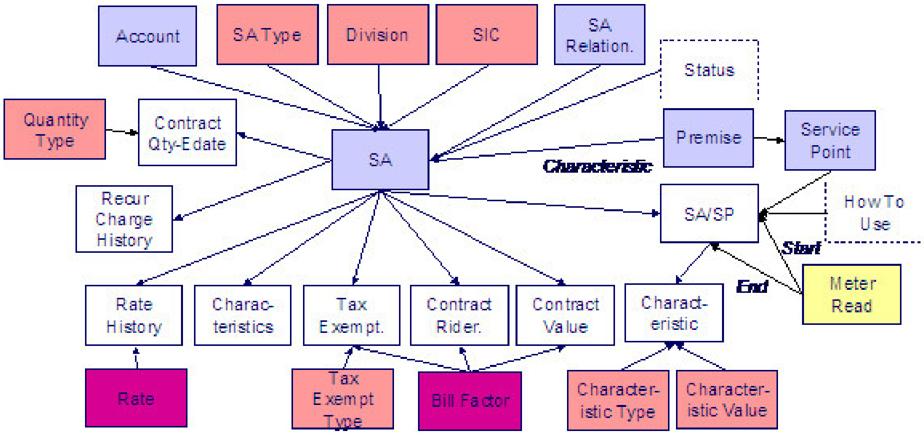
Service Point Suggestions

N/A.

Service Agreement

Service Agreement Data Model

The following data model illustrates the service agreement object.



Service Agreement Table Names

Data Model Name	Table Name	Generated Keys	Object Validation Batch Control	Referential Integrity Validation Batch Control	Key Assignment Batch Control	Insertion Batch Control
Service Agreement	CI_SA	Yes CI_SA_K	VAL-SA		CIPVSVAK Has dependencies	CIPVSVAI
Characteristic	CI_SA_CHAR	No. The key is SA_ID plus an edate and a char type.		CIPVSACV		CIPVSACI
Contract Quantity Edate	CI_SA_CONT_QTY	No. The key is service agreement ID plus quantity type plus edate.		CIPVSAQV		CIPVSAQI
Message	CI_SA_MSG	No. The key is service agreement ID plus Bill message code.		CIPVSMGV		CIPVSMGI
Recurring Charge	CI_SA_RCHG_HIST	No. The key is service agreement ID plus edate.		CIPVSARV		CIPVSARI
SA Relationship	CI_SA_REL	Yes CI_SA_REL_K	VAL-SARL	CIPVSRLV	CIPVSRLK Has dependencies	CIPVSRLI
Rate History	CI_SA_RS_HIST	No. The key is service agreement ID plus edate.		CIPVSAHV		CIPVSAHI
SA/SP	CI_SA_SP	Yes CI_SA_SP_K		CIPVSAPV	CIPVSSPK Has dependencies	CIPVSAPI
SA/SP Characteristic	CI_SA_SP_CHAR	No. The key is SA/SP Id plus char type plus effective date.		CIPVSSCV		CIPVSSCI
Tax Exempt	CI_SA_CONTERM - this table is also used for the next 2 entities, the	No. This key is service agreement ID plus CONTERM_TYPE_FLG plus		CIPVSAOV		CIPVSAOI

	key contains CONTERM_ TYPE_FLG that controls the entity	BF_CD plus START_DT		
Contract Rider	CI_SA_ CONTERM - this table is also used for the previous and next entities, the key contains CONTERM_ TYPE_FLG that controls the entity	No. This key is service agreement ID plus CONTERM_ TYPE_FLG plus BF_CD plus START_DT	CIPVSAOV	CIPVSAOI
Contract Value	CI_SA_ CONTERM - this table is also used for the previous 2 entities, the key contains CONTERM_ TYPE_FLG that controls the entity	No. This key is service agreement ID plus CONTERM_ TYPE_FLG plus BF_CD plus START_DT	CIPVSAOV	CIPVSAOI

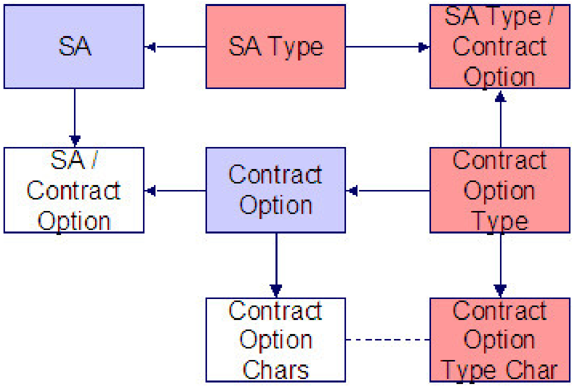
Service Agreement Suggestions

N/A.

Contract Options

Contract Options Data Model

The following data model illustrates the contract options objects.



Contract Options Table Names

Data Model Name	Table Name	Generated Keys	Object Validation Batch Control	Referential Integrity Validation Batch Control	Key Assignment Batch Control	Insertion Batch Control
Contract Option	CI_COP	Yes CI_COP_K	VAL-COP		CIPVCOPK Has dependencies	CIPVCOPI
Contract Option Language	CI_COP_L	No. The key is CONT_OPT_ID plus language code		CIPVCOLV		CIPVCOLI
Contract Option Characteristics	CI_COP_CHAR	No. The key is CONT_OPT_ID plus CHAR_TYPE_CD plus EFFDT		CIPVCCFV		CIPVCCAI
SA / Contract Option	CI_SA_COP	Yes *CI_SA_COP_K*		CIPVSCPV	CIPVSCPK Has dependencies	CIPVSCPI

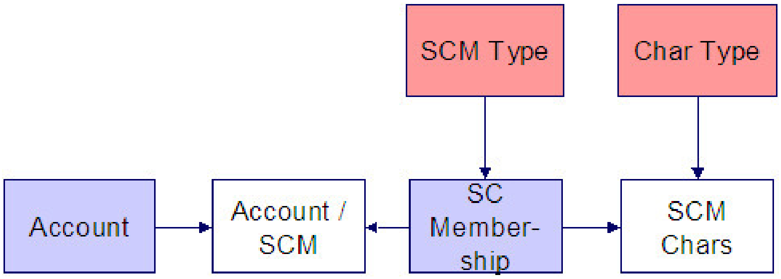
Contract Options Suggestions

N/A.

Service Credit Membership

Service Credit Membership Data Model

The following data model illustrates the Service Credit Membership objects.



Service Credit Membership Table Names

Data Model Name	Table Name	Generated Keys	Object Validation Batch Control	Referential Integrity Validation Batch Control	Key Assignment Batch Control	Insertion Batch Control
Service Credit Membership	CI_SCM	Yes CI_SCM_K	VAL-SCM	CIPVSCBV	CIPVSCMK	CIPVSCMI
Service Credit Membership / Account	*CI_SCM_ACCT*	No. The key is SCM_ID plus ACCT_ID.		CIPVSCAV		CIPVSCAI
Service Credit Membership Characteristics	CI_SCM_CHAR	No. The key is SCM_ID plus CHAR_TYPE_ CD plus EFFDT.		CIPVSCCV		CIPVSCCI

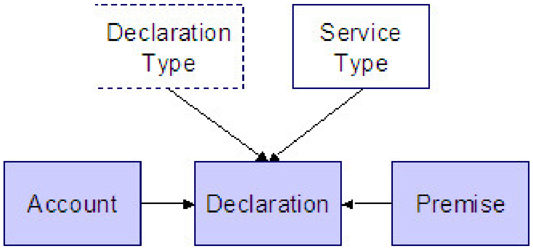
Service Credit Membership Suggestions

N/A.

Declaration

Declaration Data Model

The following data model illustrates the Declaration object.



Declaration Table Names

Data Model Name	Table Name	Generated Keys	Object Validation Batch Control	Referential Integrity	Key Assignment Batch Control	Insertion Batch Control

					Validation Batch Control		
Declaration	CI_DCL	Yes	CI_DCL_K	VAL-DCL		CIPVDCRK	CIPVDCRI

Declaration Suggestions

N/A.

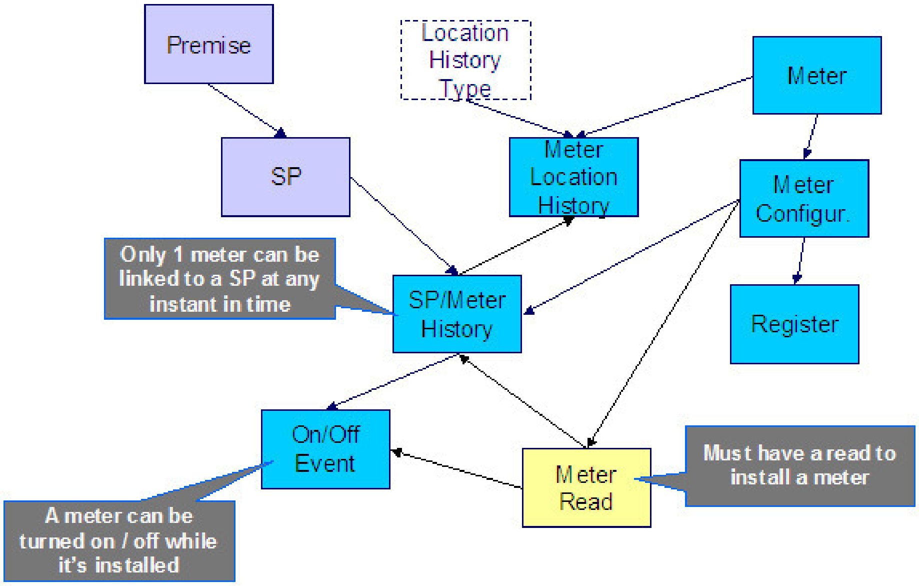
Transaction Data

This section describes the tables in which your transaction data (e.g., bills, payments, meter reads, customer contacts, etc.) resides.

SP / Meter History and On/Off Event

SP / Meter Data Model

The following data model illustrates the service point / meter installation object.



Only 1 meter can be linked to a SP at any instant in time

Must have a read to install a meter

A meter can be turned on / off while it's installed

SP / Meter Table Names

Data Model Name	Table Name	Generated Keys	Referential Integrity Validation Batch Control	Key Assignment Batch Control	Insertion Batch Control
SP/Meter History	CI_SP_MTR_HIST	Yes CI_SP_MTR_HIST_K	CIPVSMHV	CIPVSMHK Has dependencies	CIPVSMHI
On/Off Event	CI_SP_MTR_EVT	No. The key is meter history ID plus sequence number.	CIPVSMEV		CIPVSMEI
Meter Location History	CI_MTR_LOC_HIS	Yes. CI_MTR_LOC_HIS_K	CIPVMLHV	CIPVMLHK	CIPVMLHI

SP / Meter Suggestions

In order to link a meter to a service point, you must

- Link the meter's meter configuration to the service point by inserting a record on the CI_SP_MTR_HIST table.
- In addition, you must also create a CI_SP_MTR_EVT record. Note the following about this record:
  - The value of the SP_MTR_EVT_FLG should be I (for installation).
  - The value of the MTR_ON_OFF_FLG should be 1 (on).
  - It must reference a read whose read date is the installation date. The reading can be a dummy value unless this customer has not been billed since the install date (i.e., the installation has taken place recently). In this situation, this read must be the true start read for the customer. Note, this read should also be linked to the SA/SP record associated with the SA that's linked to the SP as its start read.

Bill

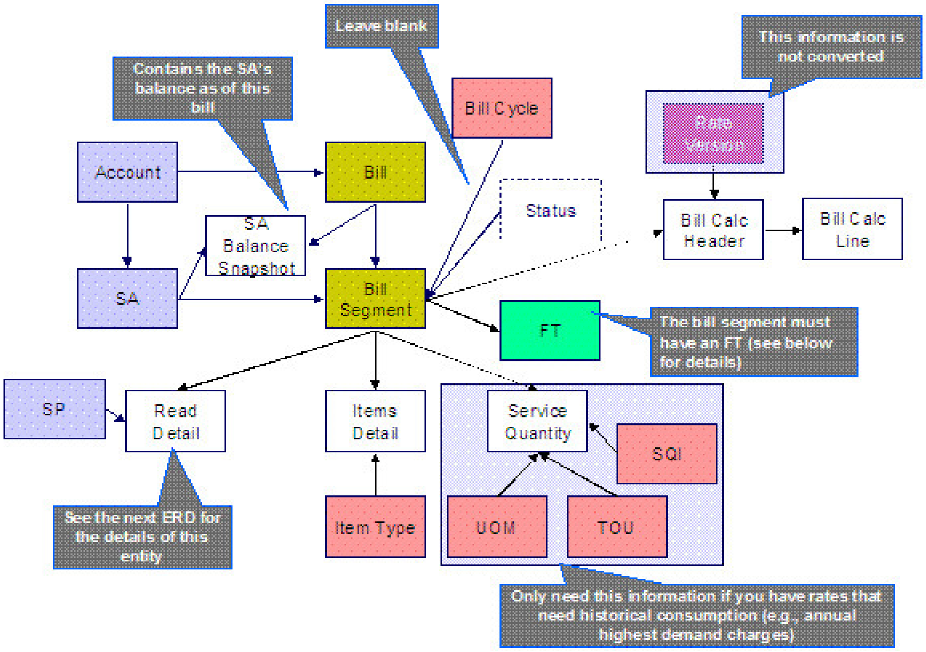
Bill Data Model

This section contains information about the Bill’s data model:

- Bill - Main
- Bill - Read Details
- Bill - FT
- Bill Characteristics
- Bill Messages, Bill Routing, and Bill Review Schedule

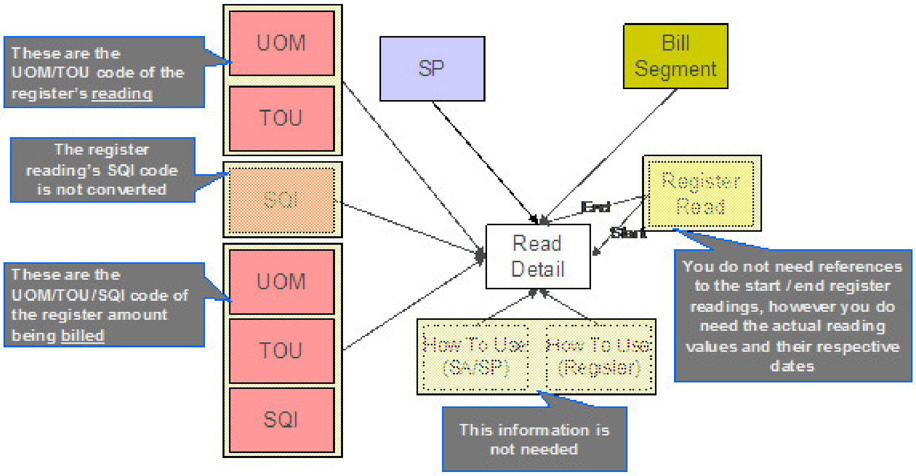
Bill - Main

The following data model illustrates the bill object.


Leave blank
This information is not converted
Contains the SA's balance as of this bill
Bill Cycle
Rate Version
Account
Bill
Status
SA Balance Snapshot
Bill Calc Header
Bill Calc Line
SA
Bill Segment
FT
The bill segment must have an FT (see below for details)
SP
Read Detail
Items Detail
Service Quantity
SQI
Item Type
UOM
TOU
See the next ERD for the details of this entity
Only need this information if you have rates that need historical consumption (e.g., annual highest demand charges)

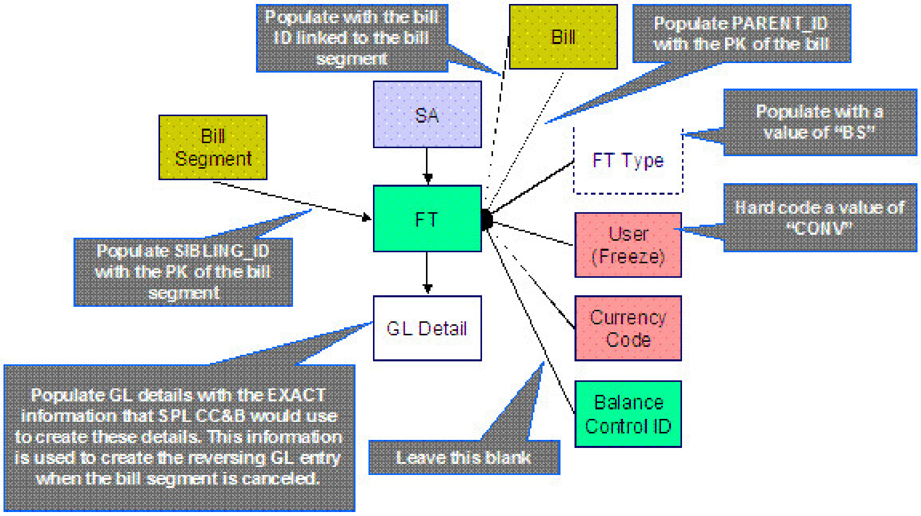
Bill - Read Details

The following data model illustrates the FK references on the read detail entity (a bill segment has one or more read details if the bill segment is associated with metered service).


These are the UOM/TOU code of the register's reading
UOM
TOU
SP
Bill Segment
The register reading's SQI code is not converted
SQI
Register Read
End
Start
Read Detail
These are the UOM/TOU/SQI code of the register amount being billed
UOM
TOU
SQI
How To Use (SA/SP)
How To Use (Register)
You do not need references to the start / end register readings, however you do need the actual reading values and their respective dates
This information is not needed

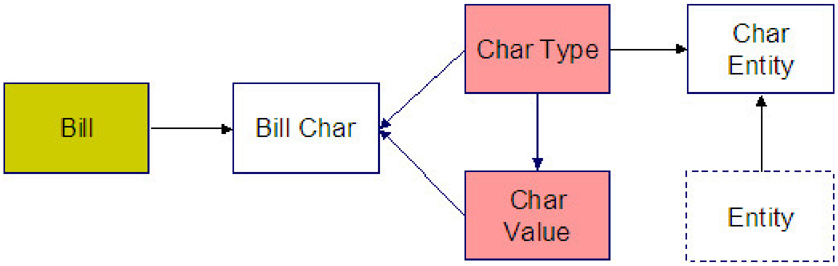
Bill - FT

The following data model illustrates the FT that must be associated with a bill segment.


Populate with the bill ID linked to the bill segment
Bill
Populate PARENT_ID with the PK of the bill
SA
Bill Segment
Populate with a value of "BS"
FT Type
FT
User (Freeze)
Hard code a value of "CONV"
Populate SIBLING_ID with the PK of the bill segment
GL Detail
Currency Code
Balance Control ID
Populate GL details with the EXACT information that SPL CC&B would use to create these details. This information is used to create the reversing GL entry when the bill segment is canceled.
Leave this blank

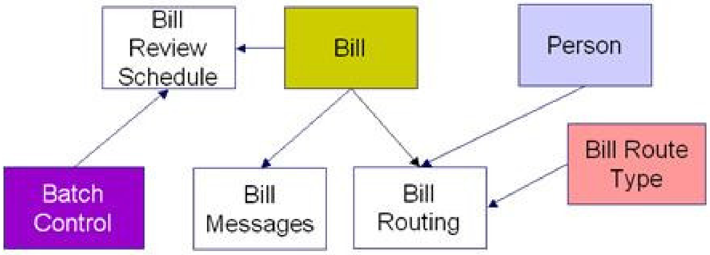
Bill Characteristics

The following data model illustrates Bill Characteristics.



Bill Messages, Bill Routing, and Bill Review Schedule

The following data model illustrates Bill Messages, Bill Routing, and Bill Review Schedule.



Bill Table Names

Data Model Name	Table Name	Generated Keys	Referential Integrity Validation Batch Control	Key Assignment Batch Control	Insertion Batch Control
Bill	CI_BILL	Yes CI_BILL_K	CIPVBLLV	CIPVBILK Has dependencies	CIPVBLLI
SA Balance Snapshot	CI_BILL_SA	No. The key is bill ID plus SA ID.	CIPVBSAV		CIPVBSAI

Bill Segment	CI_BSEG	Yes *CI_BSEG_K*	CIPVSEGV	CIPVBSGK Has dependencies	CIPVSEGI
Calc Header	CI_BSEG_CALC	No. The key is bill segment ID and a sequence number	CIPVBSCV		CIPVBSCI
Calc Lines	CI_BSEG_CALC_LN	No. The key is bill segment ID, the header sequence number, and a sequence number	CIPVBSLV		CIPVBSLI
Read Detail	CI_BSEG_READ	No. The key is bill segment ID, SP ID and a sequence number	CIPVSRRV		CIPVSRRI
Item Detail	CI_BSEG_ITEM	No. The key is bill segment id and a sequence number	CIPVBSIV		CIPVBSII
Service Quantity	CI_BSEG_SQ	No. The key is bill segment ID, uom code, tou code and SQI code	CIPVSQTV		CIPVSQTI
FT (financial transaction)	CI_FT	Yes *CI_FT_K*	CIPVFTFV	CIPVFTXK Has dependencies	CIPVFTFI
FT GL (FT general ledger)	CI_FT_GL	No. The key is FT ID and a GL sequence number	CIPVFTGV		CIPVFTGI
Characteristics	CI_BILL_CHAR	No. The key is bill ID, char type code and a sequence number	CIPVBCHV		CIPVBCHI
Bill Messages	CI_BILL_MSGS	No. The key is bill ID and bill message code.	CIPVBLMV		CIPVBLMI
Bill Routing	CI_BILL_ROUTING	No. The key is bill ID and a sequence number	CIPVBLRV		CIPVBLRI
Bill Review Schedule	CI_BILL_RVW_SCH	No. The key is bill ID, bill review date and batch code.	CIPVBRVV		CIPVBRVI

Bill Suggestions

Most companies have found it impossible to load bill segment item, bill calculation header and lines with sufficient information and therefore these tables are not populated. See the comments in the above ERD's for more information.

Please populate the columns on the FT that's associated with the bill segment as follows:

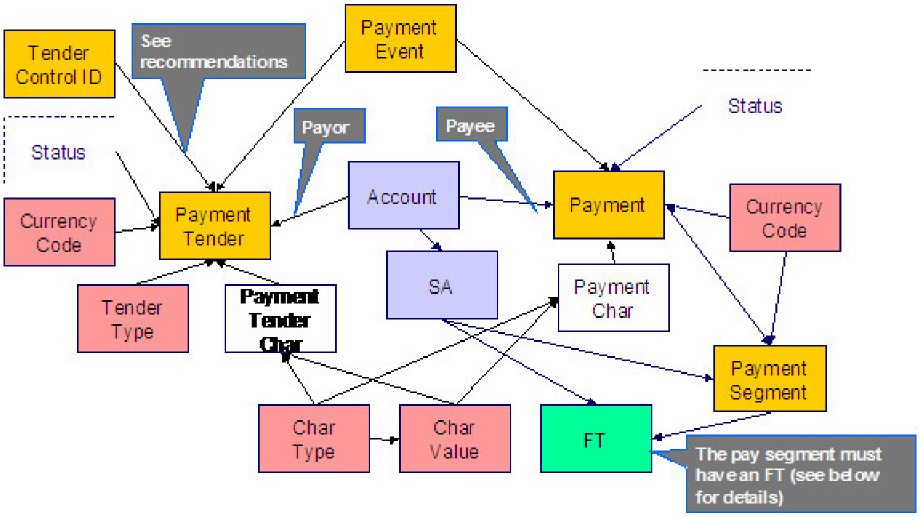
- CUR_AMT should be set equal to the bill segment amount

- PAY_AMT should be set equal to the bill segment amount
- CRE_DTTM should be set equal to the bill segment end date / time
- FREEZE_SW should be "Y"
- FREEZE_DTTM should be set equal to the bill segment end date / time
- ARS_DT should be set equal to the bill segment end date
- CORRECTION_SW should be "N"
- REDUNDANT_SW should be "N"
- NEW_DEBIT_SW should be "N"
- NOT_IN_ARS_SW should be set to "N"
- SHOW_ON_BILL_SW should be set to "N"
- ACCOUNTING_DT should be set to the current date
- SCHED_DISTRIB_DT should be left blank
- CURRENCY_CD should be the currency on the installation record
- BAL_CTL_GRP_ID should be left blank
- XFERRED_OUT_SW should be set to "Y"
- PARENT_ID should be set to the bill ID
- SIBLING_ID should be set to the bill segment ID
- Do NOT create any GL details for the FT. If GL details are converted, ensure they are populated with the EXACT information SPL CC&B would use to create them. This information is used to create the reversing GL entry when the bill segment is canceled.

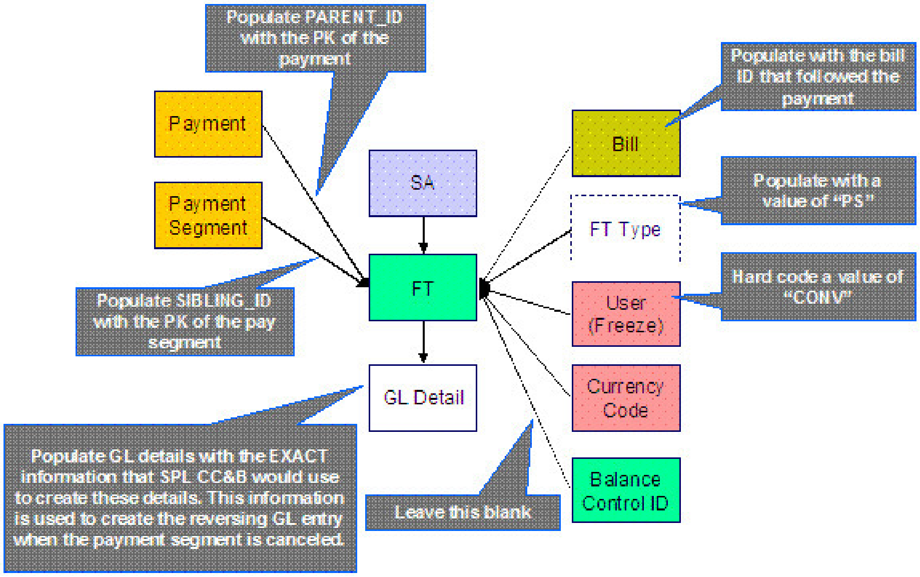
Payment

Payment Data Model

The following data model illustrates the payment object.



The following data model illustrates the FT that must be associated with a payment segment.


Populate PARENT_ID with the PK of the payment
Populate with the bill ID that followed the payment
Payment
Bill
SA
Payment Segment
Populate with a value of "PS"
FT Type
FT
Hard code a value of "CONV"
Populate SIBLING_ID with the PK of the pay segment
User (Freeze)
GL Detail
Currency Code
Populate GL details with the EXACT information that SPL CC&B would use to create these details. This information is used to create the reversing GL entry when the payment segment is canceled.
Leave this blank
Balance Control ID

Payment Table Names

Data Model Name	Table Name	Generated Keys	Referential Integrity Validation Batch Control	Key Assignment Batch Control	Insertion Batch Control
Payment	CI_PAY	Yes *CI_PAY_K*	CIPVPAYV	CIPVPAYK Has dependencies	CIPVPAYI
Payment Event	CI_PAY_EVENT	Yes CI_PAY_EVENT_K		CIPVPYEK Has dependencies	CIPVPYEI
Payment Event Characteristic	CI_PAY_EVT_CHAR	No. The key is PAY_EVENT_ID and a char type.	CIPVBLCV		CIPVBLCI

Payment Tender	CI_PAY_TNDR	Yes CI_PAY_TNDR_K	CIPVTNDV	CIPVTNDK Has dependencies	CIPVTNDI
Payment Segment	CI_PAY_SEG	Yes CI_PAY_SEG_K	CIPVPSGV	CIPVPSGK Has dependencies	CIPVPSGI
FT (financial transaction)	CI_FT	Yes *CI_FT_K*	CIPVFTFV	CIPVFTXK Has dependencies	CIPVFTFI
FT GL (FT general ledger)	CI_FT_GL	No. The key is FT id and a GL sequence number	CIPVFTGV		CIPVFTGI
Payment Tender Characteristic	CI_PAY_TNDR_CHAR	No. The key is PAY_TENDER_ID, plus a sequence number and a char type	CIPVTNCV		CIPVTNCI
Payment Characteristic	CI_PAY_CHAR	No. The key is PAY_ID, plus a sequence number and a char type	CIPVPYCV		CIPVPYCI

Payment Suggestions

We recommend that you use the system to create a single deposit control and link to it a single tender control using the PRODUCTION tables. The tender control should reference a tender source of "conversion". Use the prime key of the tender control as the foreign key on the tenders that you insert into the STAGING tables. This means you will have an invalid foreign key relationship on CI_PAY_TNDR (it will reference a tender control that doesn't exist).

After converting the payments:

- Re-access the tender control in production and enter the appropriate amounts (per tender type) to balance the tender control.
- Re-access the deposit control in production and enter the appropriate amounts to balance the deposit control.

Please populate the columns on the FT that's associated with the payment segment as follows:

- CUR_AMT should be set equal to the payment segment amount
- PAY_AMT should be set equal to the payment segment amount
- CRE_DTTM should be set equal to the payment segment date / time
- FREEZE_SW should be "Y"
- FREEZE_DTTM should be set equal to the payment segment date / time
- ARS_DT should be set equal to the payment segment date
- CORRECTION_SW should be "N"
- REDUNDANT_SW should be "N"
- NEW_DEBIT_SW should be "N"
- NOT_IN_ARS_SW should be set to "N"
- SHOW_ON_BILL_SW should be set to "N" on all payments other than payments that have been received since the last bill. For recent payments that you want to show on the next bill, this switch must be "Y"

- ACCOUNTING_DT should be set to the current date
- SCHED_DISTRIB_DT should be left blank
- CURRENCY_CD should be the currency on the installation record
- BAL_CTL_GRP_ID should be left blank
- XFERRED_OUT_SW should be set to "Y"
- PARENT_ID should be set to the payment ID
- SIBLING_ID should be set to the payment segment ID
- Do NOT create any GL details for the FT. If GL details are converted, ensure they are populated with the EXACT information SPL CC&B would use to create them. This information is used to create the reversing GL entry when the payment segment is canceled.

Adjustment

Adjustment Data Model

Adjustment Table Names

Data Model Name	Table Name	Generated Keys	Referential Integrity Validation Batch Control	Key Assignment Batch Control	Insertion Batch Control
Adjustment	CI_ADJ	Yes *CI_ADJ_K*	CIPVADJV	CIPVADJK Has dependencies	CIPVADJI
Adjustment Characteristic	CI_ADJ_CHAR	No. The key is ADJ_ID and a char type.	CIPVADCV		CIPVADCI
Adjustment A/P Request	CI_ADJ_APREQ	Yes CI_ADJ_APREQ_K	CIPVAPRV	CIPVAPRK Has dependencies	CIPVAPRI
FT (financial transaction)	CI_FT	Yes *CI_FT_K*	CIPVFTFV	CIPVFTXK Has dependencies	CIPVFTFI
FT GL (FT general ledger)	CI_FT_GL	No. The key is FT ID and a GL sequence number	CIPVFTGV		CIPVFTGI
FT Process	CI_FT_PROC	No. The key is FT id and a sequence number	CIPVFTPV		CIPVFTPI

Adjustment Suggestions

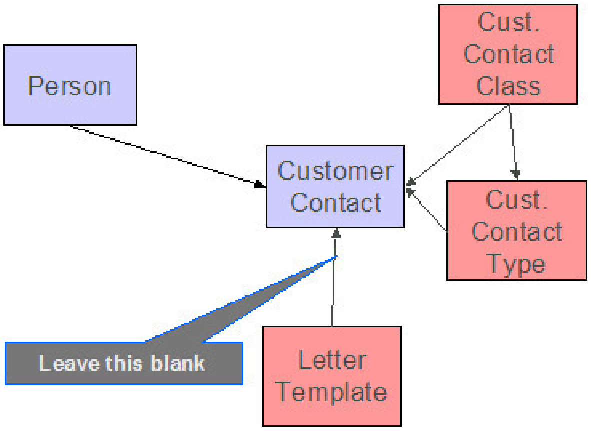
Please populate the columns on the FT that's associated with the adjustment as follows:

- CUR_AMT should be set equal to the adjustment amount
- PAY_AMT should be set equal to the adjustment amount
- CRE_DTTM should be set equal to the adjustment date / time
- FREEZE_SW should be "Y"
- FREEZE_DTTM should be set equal to the adjustment date / time
- ARS_DT should be set equal to the adjustment date
- CORRECTION_SW should be "N"
- REDUNDANT_SW should be "N"
- NEW_DEBIT_SW should be "N"
- NOT_IN_ARS_SW should be set to "N"
- SHOW_ON_BILL_SW should be set to "N" on all adjustments other than adjustments that have been generated since the last bill. For recent adjustments that you want to show on the next bill, this switch must be "Y"
- ACCOUNTING_DT should be set to the current date
- SCHED_DISTRIB_DT should be left blank
- CURRENCY_CD should be the currency on the installation record
- BAL_CTL_GRP_ID should be left blank
- XFERRED_OUT_SW should be set to "Y"
- PARENT_ID should be set to the adjustment's adjustment type
- SIBLING_ID should be set to the adjustment ID
- Do NOT create any GL details for the FT. If GL details are converted, ensure they are populated with the EXACT information SPL CC&B would use to create them. This information is used to create the reversing GL entry when the adjustment is canceled.

Customer Contact

Customer Contact Data Model

The following data model illustrates the Customer Contact object.



Customer Contact Table Names

Data Model Name	Table Name	Generated Keys	Referential Integrity Validation Batch Control	Key Assignment Batch Control	Insertion Batch Control
Customer Contact	CI_CC	Yes *CI_CC_K*	CIPVCSCV	CIPVCCTK Has dependencies	CIPVCSCI

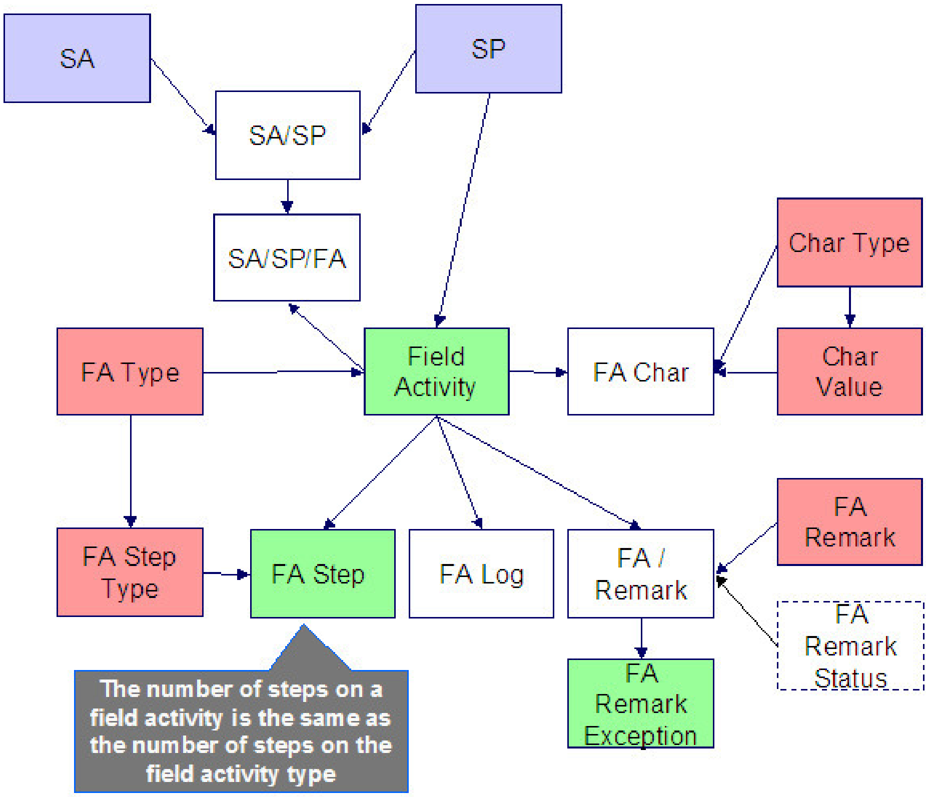
Customer Contact Suggestions

N/A

Field Activity

Field Activity Data Model

The following data model illustrates the field activity object.



Field Activity Table Names

Data Model Name	Table Name	Generated Keys	Object Validation Batch Control	Referential Integrity Validation Batch Control	Key Assignment Batch Control	Insertion Batch Control
Field Order	CI_FO	Yes *CI_FO_K*	VAL-FO	CIPVFORV	CIPVFORK Has dependencies	CIPVFORI
Field Activity	CI_FA	Yes *CI_FA_K*	VAL-FA		CIPVFACK Has dependencies	CIPVFACI
Step Not applicable to Customer to Meter	CI_FA_STEP	No. The key is FA_ID plus sequence number.		CIPVFSTV		CIPVFSTI
Characteristic	*CI_FA_CHAR*	No. The key is FA_ID plus CHAR_ TYPE_CD plus sequence number.		CIPVFAHV		CIPVFAHI

Remarks	*CI_FA_REM*	No. The key is FA_ID plus FA_ REM_CD.	CIPVFARV	CIPVFARI
Log	*CI_FA_LOG*	No. The key is FA_ID plus sequence number.	CIPVFALV	CIPVFALI
SA/SP/FA	CI_SA_SP_FA	No. The key is SA/SP id and FA id.	CIPVSSFV	CIPVSSFI

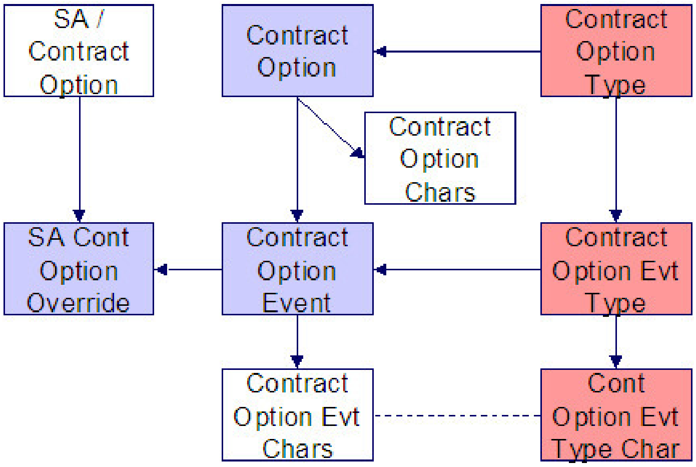
Field Activity Suggestions

N/A

Contract Option Events

Contract Option Events Data Model

The following data model illustrates the Contract Options objects.



Contract Option Events Table Names

Data Model Name	Table Name	Generated Keys	Object Validation Batch Control	Referential Integrity Validation Batch Control	Key Assignment Batch Control	Insertion Batch Control

Contract Option Event	CI_COP_EVT	Yes CI_COP_EVT_K	VAL-CEVT		CIPVCEVK Has dependencies	CIPVCEVI
Contract Option Event Characteristics	CI_COP_EVT_CHAR	No. The key is CONT_OPT_EVT_ID plus CHAR_TYPE_CD		CIPVCVCV		CIPVCVCI
SA Contract Option Override	CI_SA_COP_OVRD	No. The key is SA_CONT_OPT_ID plus CONT_OPT_EVT_ID plus OVRD_DTTM		CIPVSCOV		CIPVSCOI

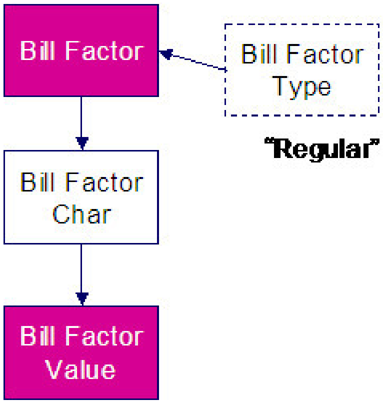
Contract Option Events Suggestions

N/A

Bill Factor Value

Bill Factor Value Data Model

The following data model illustrates the bill factor objects.



Bill Factor Value Table Names

Data Model Name	Table Name	Generated Keys	Referential Integrity Validation Batch Control	Key Assignment Batch Control	Insertion Batch Control

Bill Factor Value	CI_BF_VAL	No. The key is BF_CD plus CHAR_TYPE_CD plus CHAR_VAL plus TOU_GRP_CD plus EFFDT	CIPVBFVV		CIPVBFVI (Not threadable)

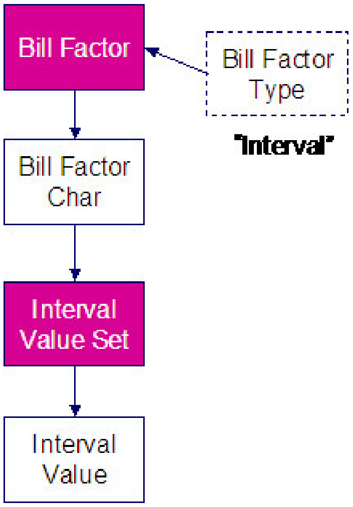
Bill Factor Value Suggestions

N/A

Bill Factor Interval Prices

Bill Factor Interval Prices Data Model

The following data model illustrates the bill factor interval prices objects.



Bill Factor Interval Prices Table Names

Data Model Name	Table Name	Generated Keys	Object Validation Batch Control	Referential Integrity Validation Batch Control	Key Assignment Batch Control	Insertion Batch Control
Interval Value Set	CI_INTV_VAL_SET	Yes CI_INTV_VAL_SET_K	VAL-IVS		CIPVIVSK	CIPVIVSI
Interval Value	CI_INTV_VAL	No. The key is INTV_VAL_SET_		CIPVITFV		CIPVITVI

		ID plus INTV_VAL_DTTM			

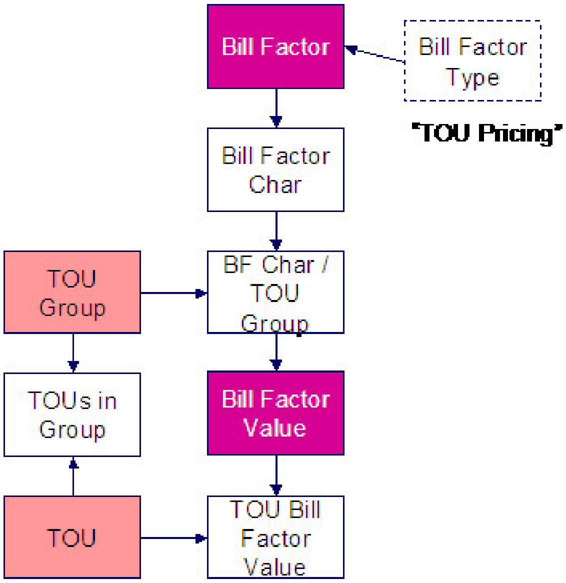
Bill Factor Interval Prices Suggestions

N/A

Bill Factor TOU Pricing

Bill Factor TOU Pricing Data Model

The following data model illustrates the bill factor TOU Pricing objects.



Bill Factor TOU Pricing Table Names

Data Model Name	Table Name	Generated Keys	Referential Integrity Validation Batch Control	Key Assignment Batch Control	Insertion Batch Control
TOU Bill Factor Value	CI_TOU_BF_VAL	No. The key is BF_CD plus CHAR_TYPE_CD plus CHAR_VAL plus EFFDT plus TOU_	CIPVTBVV		CIPVTBVI (Not threadable)

		GRP_CD plus TOU_ CD			

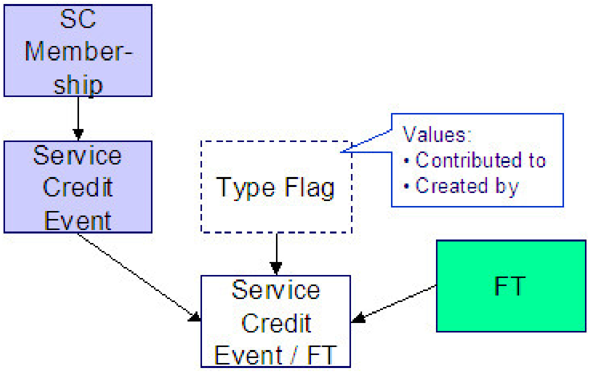
Bill Factor TOU Pricing Suggestions

N/A

Service Credit Event

Service Credit Event Data Model

The following data model illustrates the Service Credit Event objects.



Service Credit Event Table Names

Data Model Name	Table Name	Generated Keys	Referential Integrity Validation Batch Control	Key Assignment Batch Control	Insertion Batch Control
Service Credit Event	CI_SC_EVT	Yes *CI_SC_EVT_K*	CIPVSCVV	CIPVSCVK Has dependencies	CIPVSCVI
Service Credit Event / FT	CI_SC_EVT_FT	Yes CI_SC_EVT_FT_K	CIPVSCFV	CIPVSCFK Has dependencies	CIPVSCFI

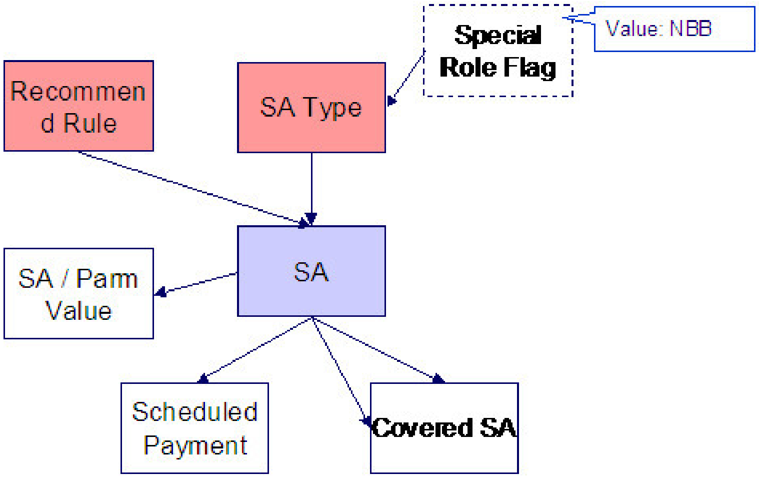
Service Credit Event Suggestions

Loading the Service Credit Event FT data is optional. This data need only be converted if it exists in the legacy system and it is deemed necessary to include it.

Non-Billed Budgets

Non-Billed Budgets Data Model

The following data model illustrates the Non-Billed Budgets objects.



Non-Billed Budgets Table Names

Data Model Name	Table Name	Generated Keys	Referential Integrity Validation Batch Control	Key Assignment Batch Control	Insertion Batch Control
NBB / Service Agreement	CI_NB_SA	No. The key is SA_ID plus CVRD_SA_ID	CIPVNBSV		CIPVNBSI
NBB Scheduled Payments	CI_NB_SCHED_PAY	Yes CI_NB_SCHED_PAY_K	CIPVNSPV	CIPVNSPK Has dependencies	CIPVNSPI
NBB / SA Parameters	CI_SA_NB_PARM	No. The key is SA_ID plus Sequence Number.	CIPVNPMV		CIPVNPMI

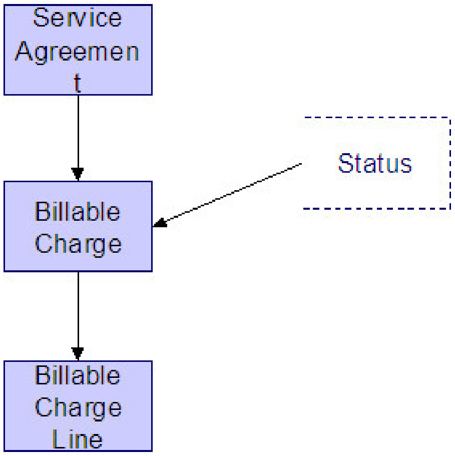
Non-Billed Budgets Suggestions

N/A

Billable Charge

Billable Charge Data Model

The following data model illustrates the Billable Charge objects.



Billable Charge Table Names

Data Model Name	Table Name	Generated Keys	Object Validation Batch Control	Referential Integrity Validation Batch Control	Key Assignment Batch Control	Insertion Batch Control
Billable Charge	CI_BILL_CHG	Yes. CI_BILL_CHG_K	VAL-BCHG	CIPVBCGV	CIPVBCGK	CIPVBCGI
Billable Charge Line	CI_B_CHG_LINE	No. The key is billable charge id and a sequence number		CIPVBCLV		CIPVBCLI

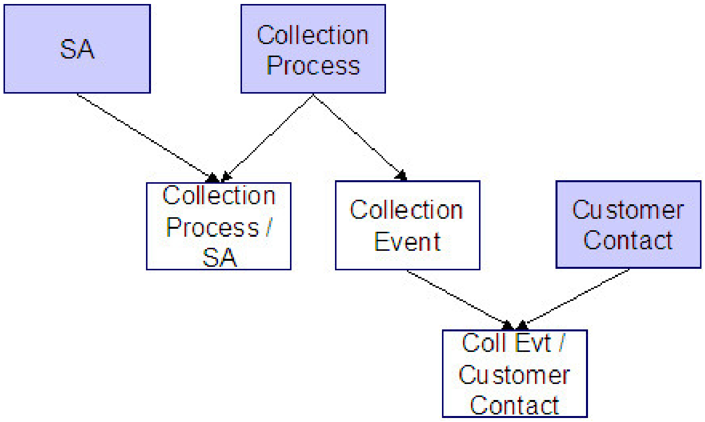
Billable Charge Suggestions

N/A

Collection Process

Collection Process Data Model

The following data model illustrates the Collection Process objects.



Collection Process Table Names

Data Model Name	Table Name	Generated Keys	Object Validation Batch Control	Referential Integrity Validation Batch Control	Key Assignment Batch Control	Insertion Batch Control
Collection Process	CI_COLL_PROC	Yes. CI_COLL_PROC_K	VAL-COLL	CIPVCLPV	CIPVCLPK	CIPVCLPI
Collection Event	CI_COLL_EVT	No. The key is collection process id and a sequence number.		CIPVCVNV		CIPVCVNI
Collection Process / Service Agreement	CI_COLL_PROC_SA	No. The key is collection process id and service agreement id.		CIPVCLSV		CIPVCLSI
Collection Event Customer Contact	CI_COLL_EVT_CC	No. The key is collection process id, a sequence number and the customer contact id.		CIPVCECV		CIPVCECI

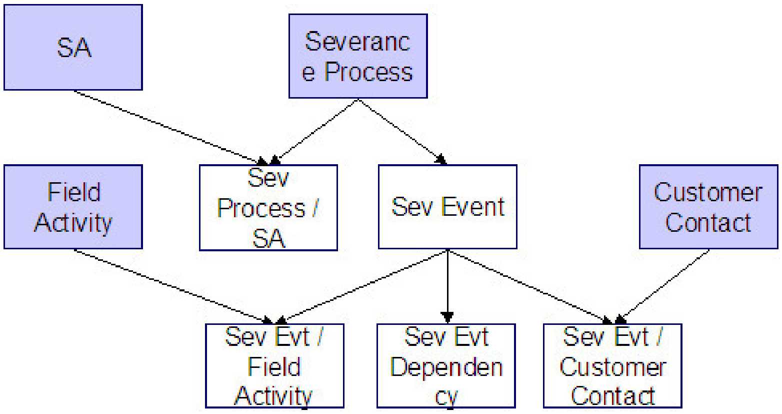
Collection Process Suggestions

N/A

Severance Process

Severance Process Data Model

The following data model illustrates the Severance Process objects.



Severance Process Table Names

Data Model Name	Table Name	Generated Keys	Object Validation Batch Control	Referential Integrity Validation Batch Control	Key Assignment Batch Control	Insertion Batch Control
Severance Process	CI_SEV_PROC	Yes. CI_SEV_PROC_K	VAL-SEVP	CIPVSEPV	CIPVSEPK	CIPVSEPI
Severance Event	CI_SEV_EVT	No. The key is severance process id and a sequence number.		CIPVSEVV		CIPVSEVI
Severance Event Customer Contact	CI_SEV_EVT_CC	No. The key is severance process id, a sequence number and the customer contact id.		CIPVSECV		CIPVSECI

Severance Event / Field Activity	CI_SEV_EVT_FA	No. The key is severance process id, a sequence number and field activity id.		CIPVSEFV		CIPVSEFI
Severance Event Dependency	CI_SEV_EVT_DEP	No. The key is severance process id, event sequence number and a sequence number.		CIPVSEDV		CIPVSEDI

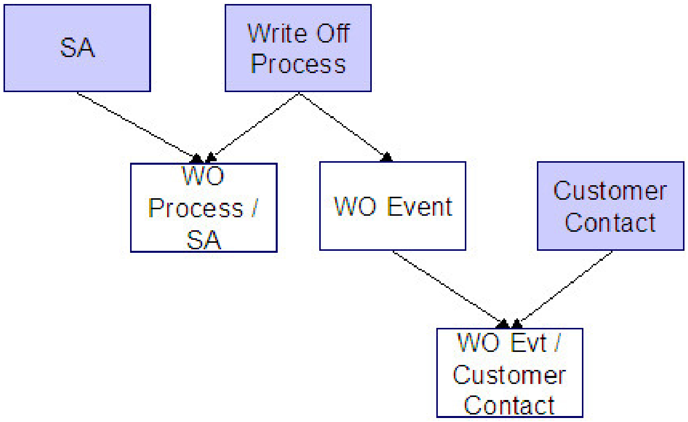
Severance Process Suggestions

N/A

Write Off Process

Write Off Process Data Model

The following data model illustrates the Write Off Process objects.



Write Off Process Table Names

Data Model Name	Table Name	Generated Keys	Object Validation Batch Control	Referential Integrity Validation Batch Control	Key Assignment Batch Control	Insertion Batch Control

Write Off Process	CI_WO_PROC	Yes. CI_WO_PROC_K	VAL-WOP	CIPVWOPV	CIPVWOPK	CIPVWOPI
Write Off Process / Service Agreement	CI_WO_PROC_SA	No. The key is write-off process id and service agreement id.		CIPVWOSV		CIPVWOSI
Write Off Event	CI_WO_EVT	No. The key is write-off process id and a sequence number.		CIPVWOVV		CIPVWOVI
Write Off Event / Customer Contact	CI_WO_EVT_CC	No. The key is write-off process id, a sequence number and the customer contact id.		CIPVWOCV		CIPVWOCI

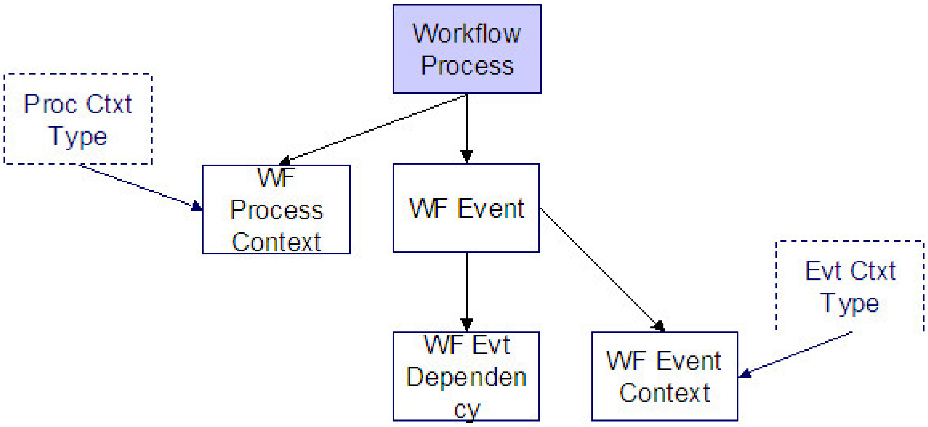
Write Off Process Suggestions

N/A

Workflow Process

Workflow Process Data Model

The following data model illustrates the Workflow Process objects.



Workflow Process Table Names

Data Model Name	Table Name	Generated Keys	Object Validation Batch Control	Referential Integrity Validation Batch Control	Key Assignment Batch Control	Insertion Batch Control
Workflow Process	CI_WF_PROC	Yes. CI_WF_PROC_K	VAL-WFP	CIPVWPRV	CIPVWPRK	CIPVWPRI
Workflow Process Context	CI_WF_PROC_CTXT	No. The key is workflow process id, workflow process context type and value.		CIPVWPCV		CIPVWPCI
Workflow Event	CI_WF_EVT	No. The key is workflow process id and a sequence number.		CIPVWEVV		CIPVWEVI
Workflow Event Context	CI_WF_EVT_CTXT	No. The key is workflow process id, event sequence number, event context type and value.		CIPVWECV		CIPVWECI
Workflow Event Dependency	CI_WF_EVT_DEP	No. The key is workflow process id, event sequence number and a sequence number.		CIPVWEDV		CIPVWEDI

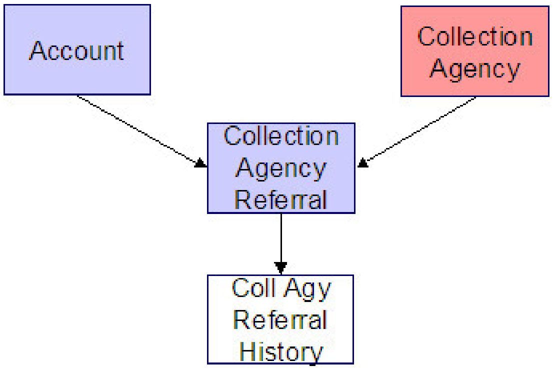
Workflow Process Suggestions

N/A

Collection Agency Referral

Collection Agency Referral Data Model

The following data model illustrates the Collection Agency Referral object.



Collection Agency Referral Table Names

Data Model Name	Table Name	Generated Keys	Referential Integrity Validation Batch Control	Key Assignment Batch Control	Insertion Batch Control
Collection Agency Referral	CI_COLL_AGY_REF	Yes. CI_COLL_AGY_REF_K	CIPVCARV	CIPVCARK	CIPVCARI
Collection Agency Referral History	CI_COLL_AGY_HIS	No. The key is collection agency referral id and characteristic type code	CIPVARHV		CIPVARHI

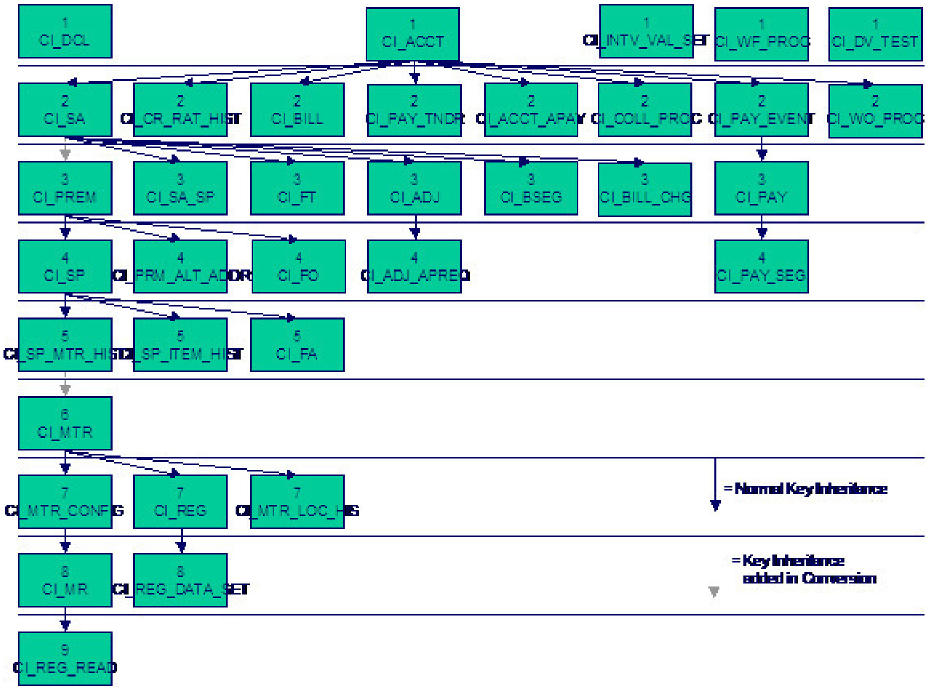
Collection Agency Referral Suggestions

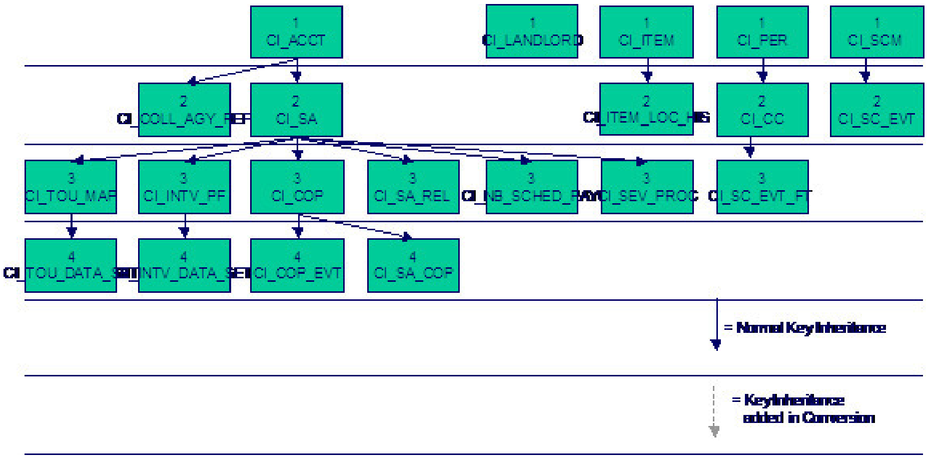
N/A

Program Dependencies

The programs used to assign production keys are listed under Master Data and Transaction Data (in the Table Names matrices). Most of these programs have no dependencies (i.e., they can be executed in any order you please). The only exceptions to this statement are illustrated in the following diagram.

The tiers in this diagram contain a box for each table whose key assignment program is dependent on the successful execution of other key assignment programs. The numbers that appear in the boxes describe the order in which these programs must be executed.



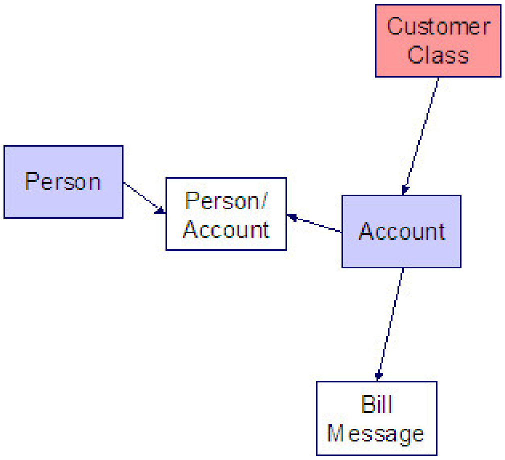


Please refer to the various "Table Names" sections above for the respective names of the programs to allocate each table's keys.

WARNING:
Prior to running the key generation program for a particular object, it is required that any previously generated keys be cleared from the key allocation tables and the key allocation temporary storage table. It is recommended that the key allocation tables be analyzed between runs to maximize performance.

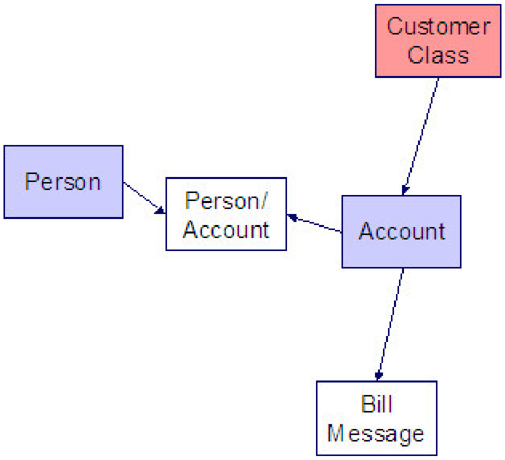
Appendix A - Entity Relationship Diagramming Standards

Because all data is stored in relational table, you need to be able to read diagrams that illustrate relationships between the various tables. The following entity diagram uses every diagramming notation used in the documentation:



Relationships

The solid line connecting the two entities that is terminated by an arrow represents a relationship between two entities. You read the relationship from the entity without the arrow to the entity with the arrow. For example in the following diagram, the line between Customer Class and Account illustrates that a Customer Class may have many Accounts, but an Account may be part of a single customer class.



Entity

Every box within the Relationships diagram represents an entity (i.e., a table). An entity may be a physical entity, such as a Person, or a logical construct, such as an Account.

Color Coding

If you can view this document in color, you will notice that each entity is colored. The color indicates the "subsystem" which governs the entity. Know the governing subsystem is important because:

- The system's menu structure is subsystem-oriented (i.e., if you know the subsystem, you will know how to use the menus to navigate to the page used to view and update the entity).
- The system's documentation is subsystem-oriented (i.e., if you know the subsystem, you will know which chapter contains information about the entity).

Some entities are not color-coded (i.e., they are white). These entities do not have a dedicated page, as they are part of a parent entity. For example, the Person / Account entity above is related to the Account object and does not have its own page. You must display the parent entity in order to view such an entity. For example, if you want to look at Person / Account information, you must go to the Account page.

The following table describes the colors utilized in the documentation:

Color	Subsystem
🟪	Customer Information
🟥	Admin (Control) Table. These tables are referenced as foreign keys on master and transaction tables. We do not document the names of these tables in this document as the table names are easily accessible using the Table transaction.
□	N/A - the entity is maintained in respect of a higher level entity.
⬚	N/A - the values in these types of entities are defined in a special table referred to as the lookup table. In order to determine the valid values for a column that references a lookup table, use the name of the column as the search value on the Look Up user interface.
🟦	Meter Management
🟨	Meter Reading
🟪	Rates
🟨	Billing
🟩	Financial Transaction
🟧	Payment
🟩	Field Order
🟪	Adjustment

Appendix B - Multiple Owners In A Single Database

In the schematic referenced in the Introduction, you'll notice that there are two table owners in the system database. We refer to the first owner as "staging" and the second owner as "production".

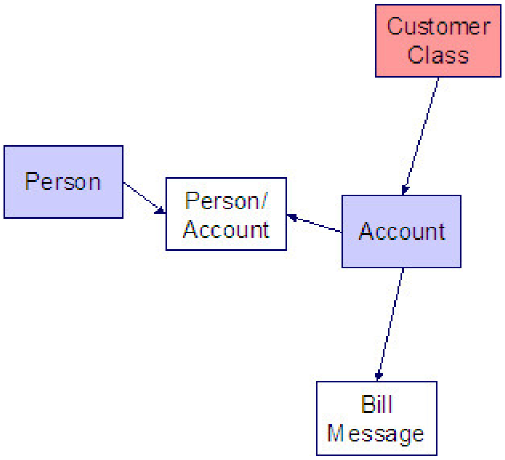
The staging owner is linked to the tables into which you insert your pre-validated data. These tables have an owner ID of CISSTG.

➤ **NOTE:**

> **Multiple staging databases.** It is possible to have multiple staging databases. In this situation, each one would have a unique owner ID, e.g., CISSTG1, CISSTG2, etc.

The production owner is linked to the tables used by your production system. These tables have an owner ID of CISADM.

When the validation programs 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 master and transaction data need to use the staging owner (CISSTG). Whereas the SQL statements that access control tables need to use the production owner (CISADM).



But notice that when these same programs run against production (Validate (b)), the SQL statements will never access the staging owner. Rather, they all point at the production owner.

This is accomplished as follows:

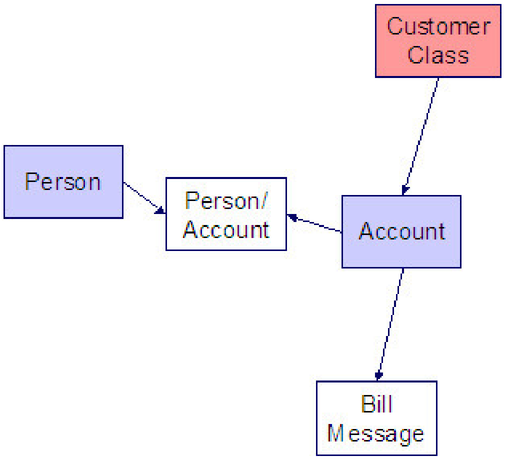
- A separate application server must exist for each owner. Each application server points at a specific database user ID.
- The database user ID associated with the staging database uses CISSTG as the owner for the master and transaction tables, but it uses CISADM as the owner of the production control tables.
- The database user ID associated with the production database uses CISADM as the owner for the master, transaction, and control 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.

While the redirection of owner ID's is a useful technique for the validation programs, it cannot be used by the key assignment and production insert programs? Why, because these programs have to access the same tables but with different owners. For example, the program that inserts rows into the person table must select rows from staging. Person and insert them into production.Person.



This is accomplished as follows:

- Views exist for each table that exists in both databases. These views have hard-coded the database owner CISADM (production). For example, there is a view called CX_PER that points at person table in production.
- The key assignment and insertion programs use these views whenever then need to access production data.

Appendix C - Known Oddities

Be aware that the following tables reference master data (e.g., persons, accounts). This means that if you look at them using a user ID that defaults ownership to the staging level, you will not be able to see the related master data (because the person / account doesn't exist in the staging owner's tables).

- Collection Agency. References a person.
- Service Provider. References a person and a service agreement.
- 3 rd Party Payor. References an account.
- Tender Source. References a suspense account.

Defining Communication Options

Device Event Types

Understanding Device Event Types

Device Events

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 which is used to. 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

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.

Refer to Understanding Device Event Types for more information.

You can access the portal from the **Admin > Communication > 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.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference D1_DVC_EVT_TYPE.

Activity Types

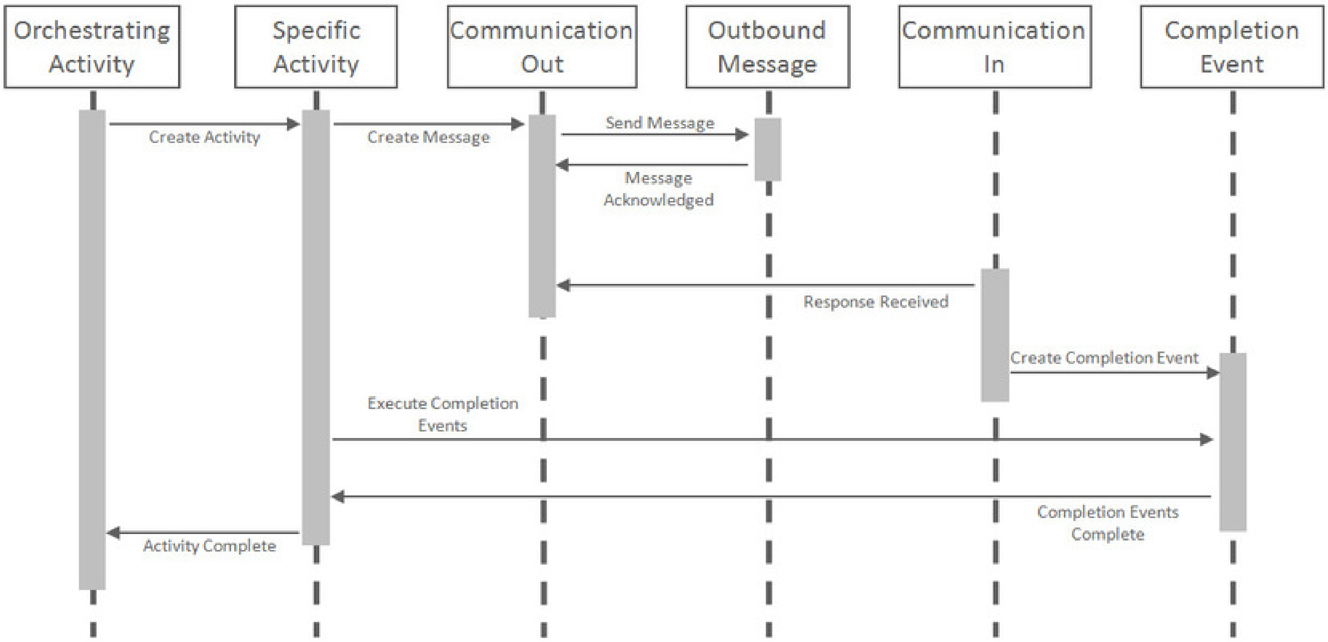
Understanding Activity Types

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

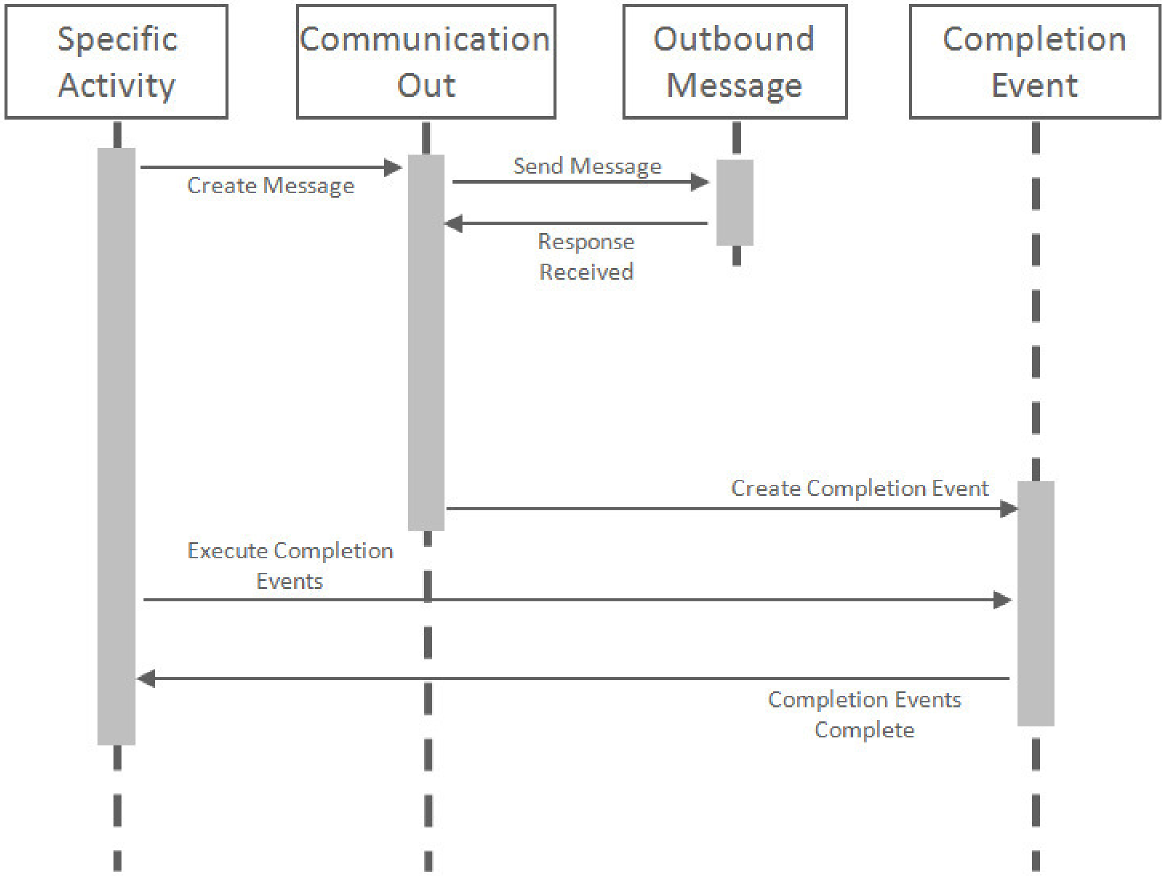
Certain types of activities such as request orchestrators, field activities, and command requests coordinate a large number of child transactions that represent the communication to and from an external application. The below diagram depicts a two way communication with an external system:



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 below 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.



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 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.

Configuring Activity Types

This portal is used to display and maintain a Activity Type.

Refer to Understanding Activity Types for more information.

You can access the portal from the **Admin** > **Communication** > **Activity Type**.

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

Where Used

Follow this link to open the data dictionary where you can view the tables that reference D1_ACTIVITY_TYPE.

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 from the **Admin > Communication > 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

Where Used

Follow this link to open the data dictionary where you can view the tables that reference D1_COMM_TYPE.

Service Task Types

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 from the **Admin** > **Customer** > **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.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference F1_SVC_TASK_TYPE.

Defining Data Extract Options

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** > **Customer 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."

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 integration section for more details on how this object is specifically put into use.

Configuring Consumption Extract Type

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 from the **Admin > Integration > 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

Where Used

Follow this link to open the data dictionary where you can view the tables that reference D1_CONS_EXT_TYPE.

Defining Aggregation Options

Configuring an Out-of-the-box Aggregation

Aggregation calculations should be run on an as needed basis. This can include running the following batches:

- 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 Data Management/Smart Grid Gateway Business User Guide* for more information on this functionality. Refer to Aggregation 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) Maintenance: D2-AggMCMaint (Service Type and Postal Aggregator-Maintenance)
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) 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 Data Management / Smart Grid Gateway 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 Data Management / Smart Grid Gateway 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 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 batch control called **Aggregation Dimension Scanner Monitor** (D1-ADS).

Defining Consumption Synchronization Options

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 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 MCs 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.

➤ **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.
- **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
D1-UPD-DTMC	Update Latest Measurement Date/Time on MC with Consumption Sync	D1-InitialLoadIMDInterval D1-ManualIMDInterval
D1-UDTSCMCRE	Update Latest Measurement Date/Time on Scalar MC with Consumption Sync	D1-InitialLoadIMDScalar 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	D1-RelMCREScalar

	on scalar measuring components. One estimation initial measurement will be created per measurement needing to be re-evaluated.	
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

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.

Defining Dashboard Options

Configuring the MDM Operational Dashboard

This section describes the process for configuring the MDM Operational Dashboard.

Refer to the Using the MDM Operational Dashboard section of the Oracle Utilities Meter Data Management/Smart Grid Gateway *Business User Guide* for a information on this functionality.

Configuration for the MDM Operational Dashboard 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, configure both the **MDM Operational Dashboard Configuration** record. 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.

This dashboard also leverages a method of pre-staging data known as Statistics Snapshots. Refer to the *Framework Administrative User Guide* for more information on Statistics Snapshots. For additional information see About Statistics in the Oracle Utilities Application Framework *Administrative User Guide*.

A method for tracking Performance Targets (also known as Service Level Agreements) is available within the MDM Operational Dashboard as well. To understand how to set up Performance Targets, refer to About Performance Targets in the Oracle Utilities Application Framework *Administrative User Guide*. The Performance Targets you configure will then be leveraged on the **Batch Performance** and/or **Database** tabs of the MDM Operational Dashboard.

Configuring the Service Order Operational Dashboard

This section describes the process for configuring the Service Order Operational Dashboard.

Refer to the Using the Service Order Operational Dashboard section of the *Oracle Utilities Meter Data Management / Smart Grid Gateway Business User Guide* for more functional information.

Configuration for the Service Order Operational Dashboard 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 **Service Order Management Master Configuration** record. 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 **Chart Options** section is the primary area for adding configuration that will affect the dashboard. Use the embedded help to guide you through the meaning of each configuration field.

Configuring the Service Order Trends Dashboard

This section describes the process for configuring the Service Order Trends Dashboard.

Refer to the Using the Service Order Trends Dashboard section of the *Oracle Utilities Meter Data Management / Smart Grid Gateway Business User Guide* for more functional information.

Configuration for the Service Order Trends Dashboard 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 **Service Order Management Master Configuration** record. 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 **Chart Options** section is the primary area for adding configuration that will affect the dashboard. Use the embedded help to guide you through the meaning of each configuration field.

Defining Measurement Reprocessing Options

Configuring Measurement Reprocessing

This section describes the process for configuring Measurement Reprocessing.

Refer to the About Measurement Reprocessing section of the *Oracle Utilities Meter Data Management / Smart Grid Gateway 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 from the **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 changes or Install Event installation constant. If either of these attributes change for a device, then a new Activity will be created that attempts to reprocess the measurements for the affected period.

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 retries any Measurement Reprocessing activities that fail to the Issue Detected state.

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
- 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 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
• 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 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
Device Event	Activity	Create date Superseded by paired event create date if earlier (where applicable)	Related paired events must be eligible Related service issue monitor must be complete Related activity must be complete	
Activity	Parent activity	Create date Superseded Device event, child activity, or completion event create date supersede (where applicable)	Cascaded transactions must be eligible Command requests must not be associated to a non-final service issue monitor Command requests must not be associated to an install event on off history entry	Child activities Device events Outbound communications Inbound communications Completion events
Outbound Communication	Activity	Create Date		Inbound communication

		Superseded by initiating activity's create date (where applicable)	
Inbound Communication	Outbound Communication	Create date Superseded by initiating outbound communication create date (where applicable)	Device events

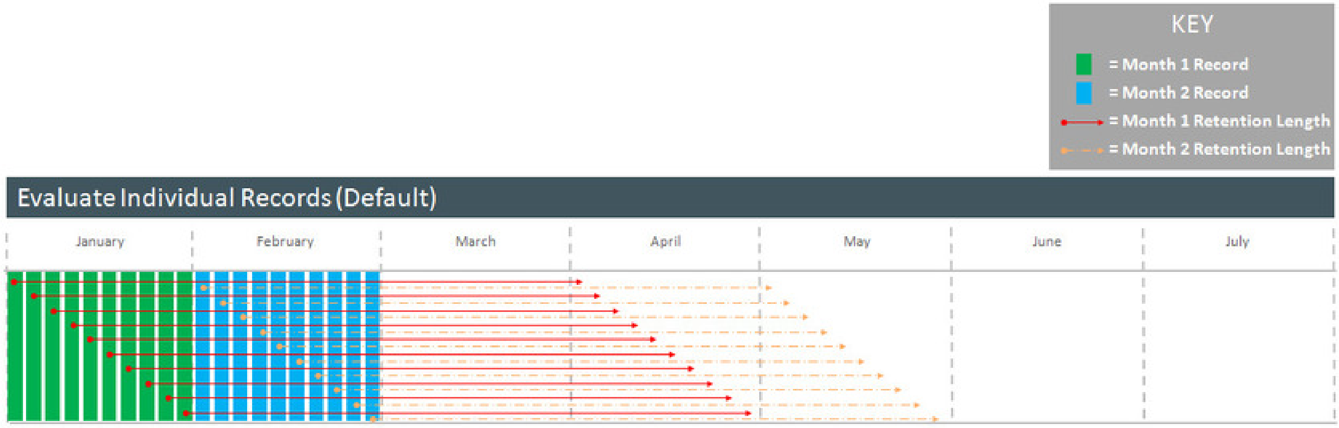
Multiple Retention Period Eligibility Crawling Strategies

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.

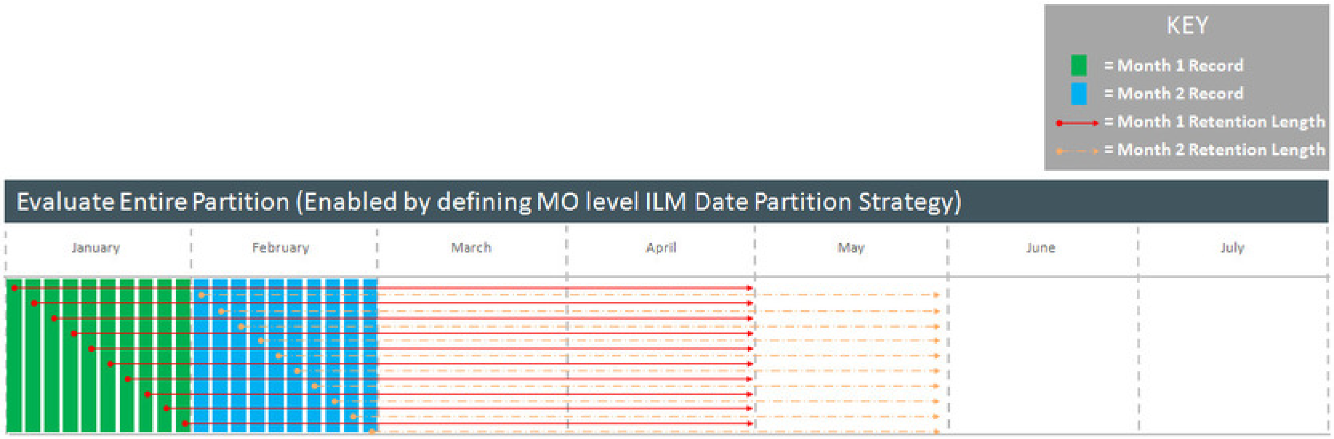
The system supports two methodologies for doing so:

- Evaluate Individual Records:
- This is our default approach
- This is the standard approach to evaluating ILM retention for all maintenance objects with a single retention period
- 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 we will process a portion of a partition each day and only once we had processed the entire partition would the DBA be able to take an archiving action on 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 had expired. All records would then be archived together when the last day of the partition was marked as eligible. This results in daily processing of small amounts of records but frequent visits to the partition.


KEY
🟩 = Month 1 Record
🟦 = Month 2 Record
= Month 1 Retention Length
= Month 2 Retention Length
Evaluate Individual Records (Default)
January
February
March
April
May
June
July

- Evaluate the Entire Partition

- This option will be enabled by configuring the ILM Date Partition Months in the maintenance object options
- In this option all records on a 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 we would not evaluate any records in the January partition until January 31st had reached the end of its retention period.
- This means that we would evaluate the 1st through the 31st all at one time
- This option should typically result 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.


KEY
= Month 1 Record
= Month 2 Record
= Month 1 Retention Length
= Month 2 Retention Length
Evaluate Entire Partition (Enabled by defining MO level ILM Date Partition Strategy)
January
February
March
April
May
June
July

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 the **Admin > General > 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.
5. **ILM Eligibility - Device Event** (D1-ILMELGDE): determines eligibility of Initial Measurement data and cascades VEE Exceptions.
6. **ILM Eligibility - Usage Transaction** (D1-ILMELIGUT): determines eligibility of Usage Transactions.
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.
- Payload Statistics Activities: Payload Statistics activities can either be the Payload Summary record or the Payload Statistic record. Eligibility is based on the Payload Statistics activity, which is only be eligible for archiving if all the related Payload Summary and Payload Error activities are also eligible for archiving. These activities are only archived by themselves if they do not have a related activity. Otherwise both the payload statistics and the payload summary activities are archived at the same time.
- 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*.

Defining Outage Storm Mode Options

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 Data Management/ Smart Grid Gateway 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 Configuratio**

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:
        a. Read Percentage
        b. Count of Received Measuring Components
        c. 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 Type**. 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**

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.

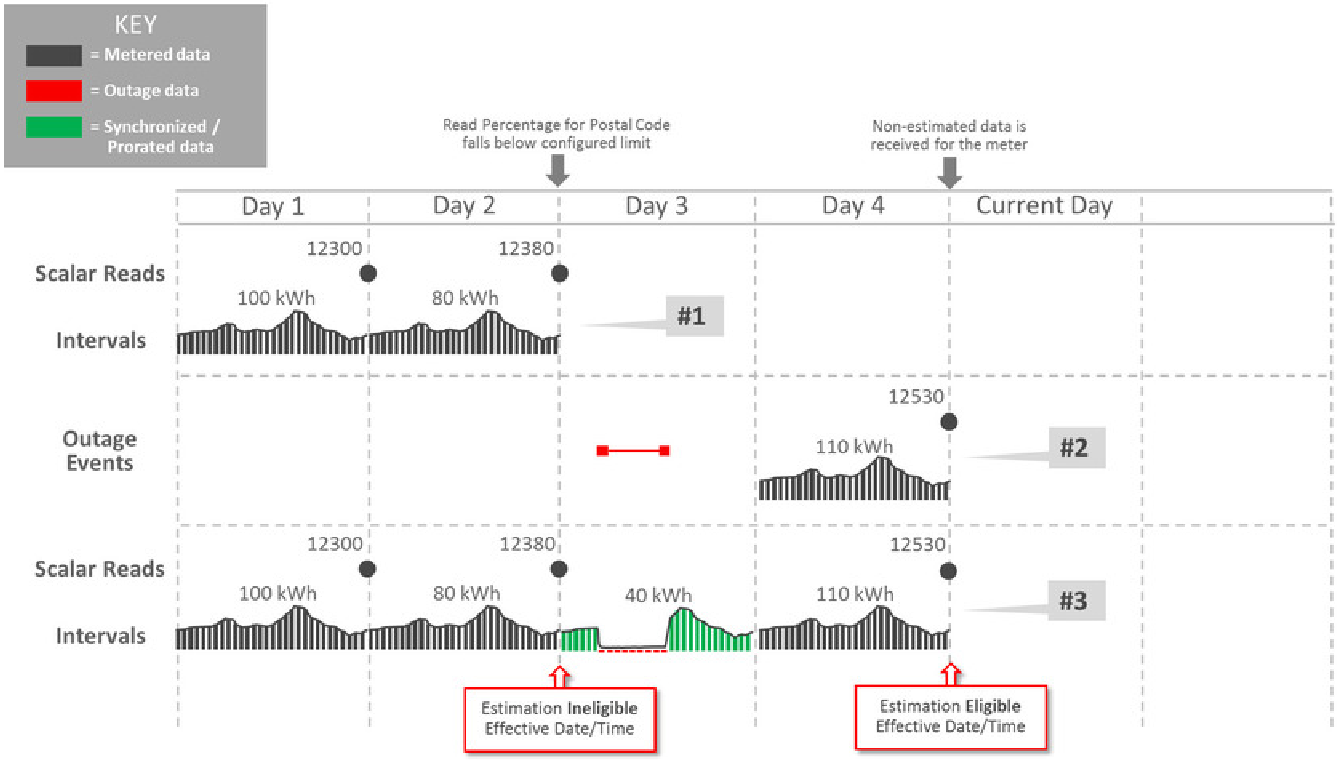
Detailed Examples of Outage Storm Mode

Scenario 1: Meter is marked estimation ineligible then received metered data 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:


KEY
= Metered data
= Outage data
= Synchronized / Prorated data
Read Percentage for Postal Code falls below configured limit
Non-estimated data is received for the meter
Day 1
Day 2
Day 3
Day 4
Current Day
Scalar Reads
Intervals
12300
12380
100 kWh
80 kWh
#1
Outage Events
12530
110 kWh
#2
Scalar Reads
Intervals
12300
12380
12530
100 kWh
80 kWh
40 kWh
110 kWh
#3
Estimation Ineligible Effective Date/Time
Estimation Eligible Effective Date/Time

Explanation for diagram:

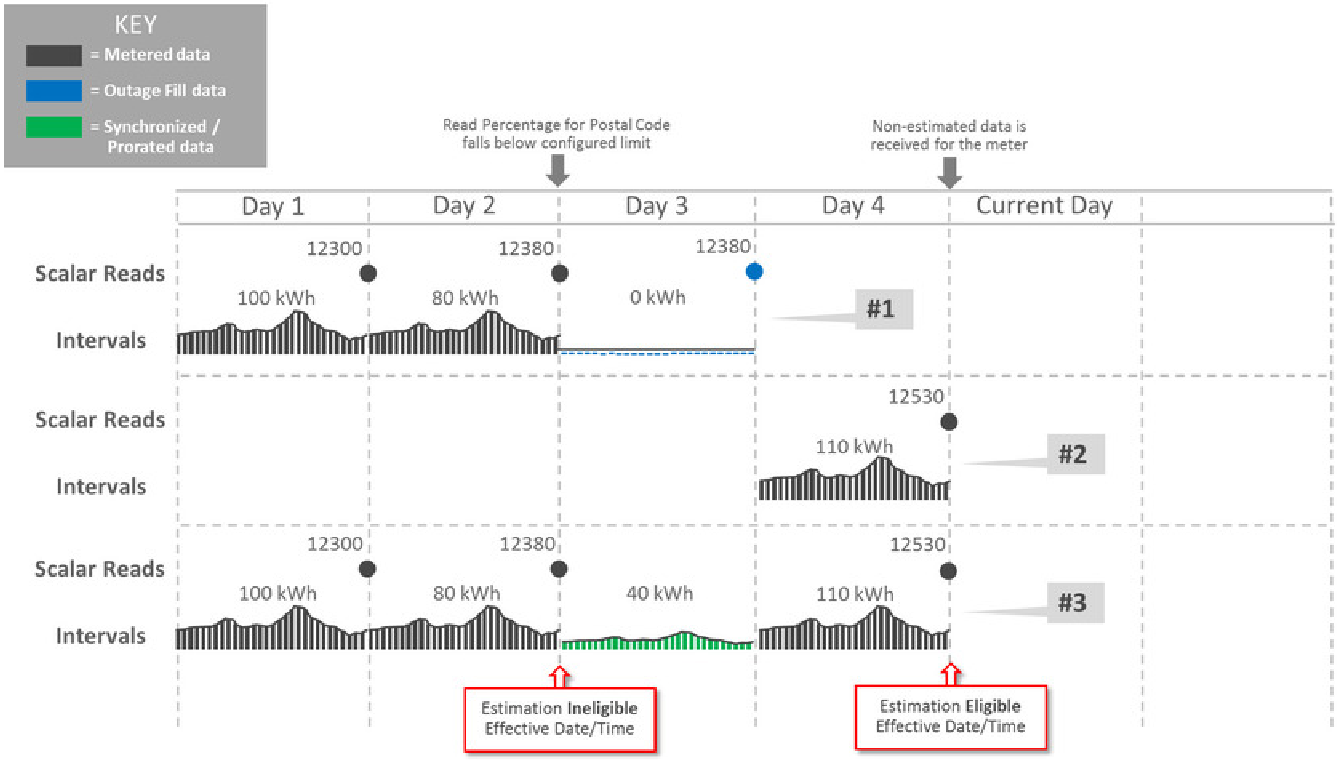
1. The meter was communicating consistently but then a widespread outage occurs in the same postal code so the meter has an estimation suppression created.
2. Two events are received...a Power Down event and a Power Up event. Also, Regular usage data is received from the meter. The reception of non-estimated 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 D1-SMMTR 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:



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.