

**Oracle Utilities Customer Care and Billing**

Business Process Guide

Volume 3

Release 2.3.1

**E18367-01**

September 2010

Oracle Utilities Customer Care and Billing Business Process Guide

E18367-01

Copyright © 2000, 2010, Oracle and/or its affiliates. All rights reserved.

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

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

If this software or related documentation is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

**U.S. GOVERNMENT RIGHTS**

Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle America, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

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

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

This software or hardware and documentation may provide access to or information on content, products and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third party content, products and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third party content, products or services.

# Rates

When you set up a customer in the system, you create a service agreement for every service that you sell to the customer. The most important information specified on a service agreement is the rate. The rate controls:

- How the service's charges are calculated.
- How the charges appear aesthetically on the customer's bills.
- How the general ledger is affected by the charges.

In addition to calculating charges for service, your implementation can also use rates to calculate the value of [adjustments](#).

In this section, we explain how to set up your organization's rates.

Refer to [How Rates Affect The Information On A Bill](#) for how billing uses the information in a rate to calculate a bill segment. Refer to [Calculated Adjustments](#) for how to use a rate to calculate an adjustment.

**Not all service agreements reference a rate.** It's important to understand that you may have service agreements that don't use a rate to calculate their charges. For example, you may have [billable charge](#) service agreements that exist to bill a customer for charges that have been calculated in a foreign system. Whether or not a service agreement uses a rate and the list of rates that can be specified on a service agreement are controlled by the service agreement's [SA type](#).

**Warning!** Do not equate a rate in your legacy system with a rate in this system. You may find that the system's flexibility will result in fewer rates than you have today. For example, your current rate structure may have 3 rates for water service because there are different prices depending on the geographic area in which the customer lives. However in this system, you can set up a single rate and vary the price depending on where the customer lives. Typically, you will only gain the intuitive understanding to make these types of decisions after attempting to set up your rates for the first time. In other words, the rate definition process is iterative and you may discover in later iterations that you can consolidate several existing rates into one flexible rate.

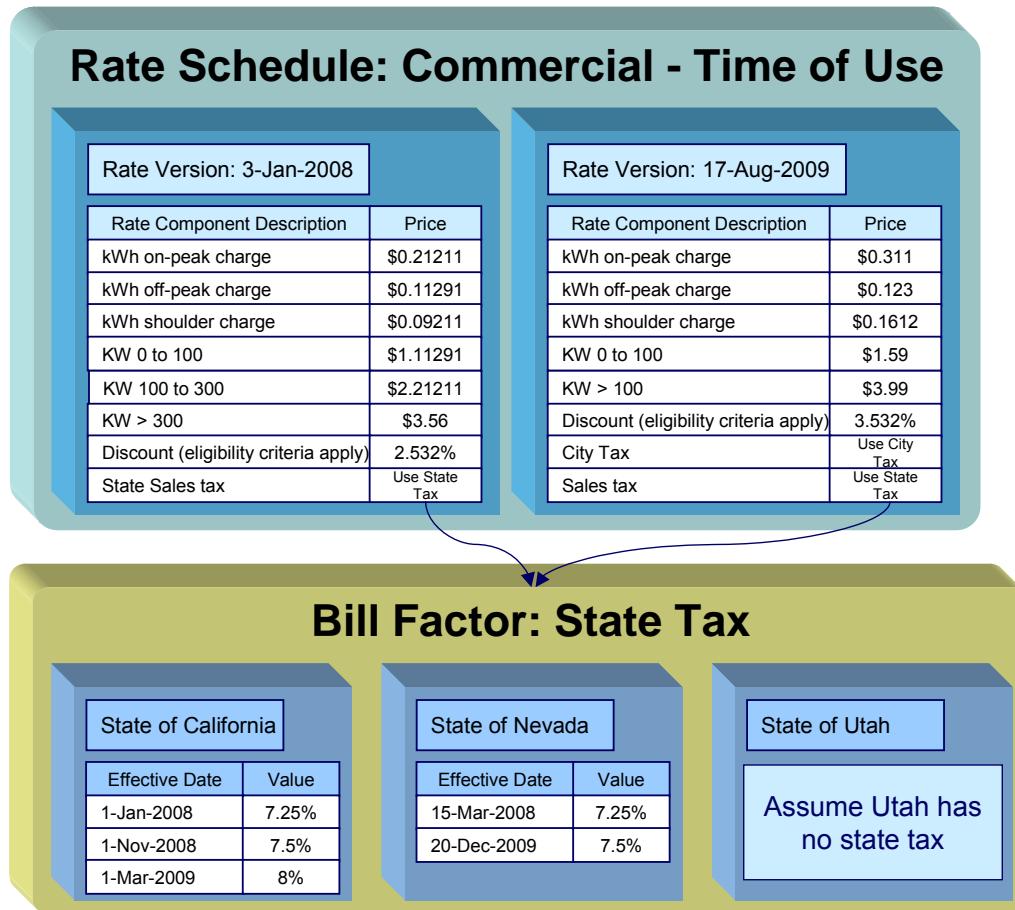
## Contents

- All Rates Share A Common Structure
- How To Create A New Rate
- Control Tables That Must Be Set Up Before Creating A Rate Schedule
- Setting Up A Rate Schedule
- Rate Schedule Merge
- Setting Up Bill Factors
- Defining Rate Versions
- Designing Rate Components
- Defining Rate Components
- Rate Version Merge
- Rate Check
- Use Sub Rate Schedules To Encapsulate Common Rate Components

[Setting Up Estimated Consumption For Items](#)  
[Setting Up Degree Days \(and Average Temperatures\)](#)  
[Effective Dates & Price Proration](#)  
[Rate Examples](#)  
[How To Add A New Service Quantity Rule Algorithm](#)  
[How To Add A New Register Rule Algorithm](#)

## All Rates Share A Common Structure

All rates share a common structure as illustrated in the following sample rate:



This rate is constructed using the following components:

- Every rate has a single **rate schedule** that contains information about the rate that doesn't change over time. For example, the rate's description.
- A rate's effective-dated calculation algorithms are stored in a **rate version**. Every rate schedule has at least one rate version. Multiple rate versions exist when something about the algorithms changes and it's important to keep the prior version in order to recalculate historical bills. The rate shown above has 2 rate versions; one effective on 3-January-2008, the other effective on 17-August-2009.

- A rate version's calculation algorithms are stored in **rate components**. Every rate version will have at least one rate component. The number of rate components linked to a rate version is dependent on the complexity of the calculation rules. We have seen very simple calculation rules that need only a single rate component; we have seen others that require over 150 rate components. You will find the construction of rate components to be the most challenging task of the setup process.
- **Bill factors** are used to specify the amount to charge when the amount is the same for many rates. Notice that both of the above rate versions reference the prevailing "state tax" bill factor rather than specifying the specific tax percent. This allows the tax rate to be maintained in one place.

**Bill factors have many uses.** Bill factors have many uses in addition to specifying tax rates. Refer to [An Overview Of Bill Factors](#) for all the details.

The remainder of this chapter discusses the intricacies of setting up the rates and bill factors.

## How To Create A New Rate

---

The following points describe steps you'll follow to set up a rate.

- Create a rate schedule for the rate. Refer to [Setting Up A Rate Schedule](#) for more information.
- Set up a rate version to establish the earliest effective date of the rate. Refer to [Defining Rate Versions](#) for more information. We recommend that you set the status of the rate version to **In Progress** while you're working on the rate.
- Set up rate components to define the rate version's calculation rules. Refer to [Defining Rate Components](#) and [Rate Version Merge](#) for more information.
- After all rate components have been setup, return to [Rate Version - Main](#) and change the rate version's status to **Validated**. At this point, you can use [Rate Check](#) to test the rate version.
- After you have checked the rate, return to the [Rate Version - Main](#) and change the status to **Finished**. This allows the rate to be used by billing.
- Define the various SA types that can use the rate. You do this by referencing the rate on [SA Type – Rate](#). Note that only primary rate schedules may be referenced on SA Types.

## Control Tables That Must Be Set Up Before Creating A Rate Schedule

---

This section describes tables that must be set up before you can add a new rate schedule.

### Contents

- [Defining Frequency Codes](#)
- [UOM versus TOU versus SQI](#)
- [Defining Measured Quantity Manipulation Rules](#)
- [Defining Service Quantity \(SQ\) Identifiers](#)

## Defining Frequency Codes

When you create a rate schedule, you must enter a code defining the frequency in which the prices are expressed, e.g., monthly, quarterly, biannual, etc. To set up frequency codes, open **Admin Menu, Frequency**.

### Description of Page

Enter a unique **Frequency ID** and **Description** for each frequency.

Before entering the remaining fields, you must analyze the “proration provisions” in your rates. For example, consider a rate used to calculate charges for a “month’s worth of service”. Assume this rate’s charges will not be prorated when the customer’s service period is anywhere between 27 and 33 days. This rate requires a frequency that indicates:

- **12 Periods / Year** (because it’s a monthly rate).
- **Minimum Days Offset** equal to **3**.
- **Maximum Days Offset** equal to **3**.

When the system creates a bill using a rate with this frequency, it compares the number of days of consumption on the bill to the minimum and maximum number of days inferred by the rate’s frequency. In the example above, the minimum number of days would be 27 (365 days / 12 periods per year - 3 offset for minimum days). The maximum number of days would be 33 (365 days / 12 periods per year + 3 offset for maximum days). Therefore, if a bill’s consumption period is less than 27 days or more than 33 days, the bill’s charges are prorated.

**Overriding the system’s standard proration logic.** If the standard proration logic does not satisfy your requirements, you may plug in an override proration algorithm to calculate the proration factors as required by your implementation. For example, we have encountered companies who treat a year has have 360 days rather than 365 days. Such a company would need to develop an override proration algorithm and plug it on the [Installation](#) record.

### Where Used

Every primary Rate Schedule must have a frequency. Refer to [Rate Schedule – Main](#) for more information.

## UOM versus TOU versus SQI

Almost all rates contain prices in respect of some number of something. This something could be anything from a therm of gas to a kilowatt-hour of electricity. We use three distinct codes to identify the “thing” being priced in a rate. These codes are **UOM** (unit of measure), **TOU** (time of use), and **SQI** (service quantity identifier). These codes must be set up before you can create a rate schedule.

You don’t have to use all three codes when you specify a price in a rate. Rather, you use the codes that are needed to clearly identify the thing being priced. We’ll use a few examples to clarify this concept.

Let’s consider a rate that simply has a price per therm. For this rate, you’d only use a UOM code to identify the thing being priced (and this code should be something simple like **THRM**).

Next, let's consider a rate that has a different price for kilowatts consumed during peak hours as compared to those used during off-peak hours. You would use a combination of UOM and TOU to define the thing being priced. In this example, we'd recommend using a single UOM for kilowatt-hours (labeled **KWH**) and two TOU codes – one for the peak period (labeled **PEAK**) and the other for the off-peak period (labeled **OFFPEAK**).

**Rule.** If the thing being priced is measured by a meter, you must use the same UOM / TOU code(s) in the rate as are specified on the meter's register(s). Refer to [Setting Up Unit Of Measure Codes](#) for more information.

And finally, let's consider a rate that has a charge for "annual highest KW (demand) during peak hours". This is a very complicated thing to price. You have several options in respect of the combination of UOM, TOU, SQI to use to express this unit. The following table contains every conceivable combination to express this single thing:

Option	UOM	TOU	SQI
1	KW	PEAK	ANNHIGH - Annual Highest
2	KW		PEAKANHI - Peak Annual Highest
3			ANHIPKKW - Annual Highest Peak KW
4	KWAP - KW Annual Highest Peak		
5	KW	AHPEAK – Annual Highest Peak	

**Rule Of Thumb.** We strongly recommend using an SQI to identify things not measured by your meters. This means that you shouldn't use the last two options in the above table. Of the first 3 options, the choice is yours. It's really a question of how you prefer to record your billed consumption.

For more information about UOM, TOU, and SQI, refer to [Setting Up Unit Of Measure Codes](#), [Setting Up Time Of Use Codes](#), and [Defining Service Quantity \(SQ\) Identifiers](#). For more information about charging for units that are not measured by your meters, refer to [Defining Measured Quantity Manipulation Rules](#).

## Defining Measured Quantity Manipulation Rules

This section describes when and how you have the system manipulate measured consumption before it is applied to the charges in a rate.

**Warning!** Quantity manipulation rules exist to support calculations performed by rate components. This means that you must design your rate components before you can design your quantity manipulation rules. We strongly recommend that you design "on paper" how every rate's rate components looks before you attempt to set up these rules.

**Contents**

- [When And How To Manipulate Measured Consumption](#)
- [Setting Up Register Rules](#)
- [Setting Up Service Quantity \(SQ\) Rules](#)

## When And How To Manipulate Measured Consumption

Many rates calculate charges based on measured consumption. For example, a simple electric rate has a price expressed as an amount per kilowatt-hour (kWh) and the customer's meter measures kWh. However, other rates contain charges for things that aren't measured directly by the customer's meters. Examples include the following:

- Most gas meters measure consumption in cubic feet, but the charges in gas rates are expressed in therms.
- A customer's service agreement may indicate that the number of kilowatts billed per month will be the greater of the measured kilowatts OR the minimum number of kilowatts as defined in the service agreement.
- A customer's meter may only contain two registers: one that measures kWh consumed during peak hours, and another that measures kWh consumed during off peak hours. However, there are charges in the rate based on the total kWh consumed regardless of when the power was used.

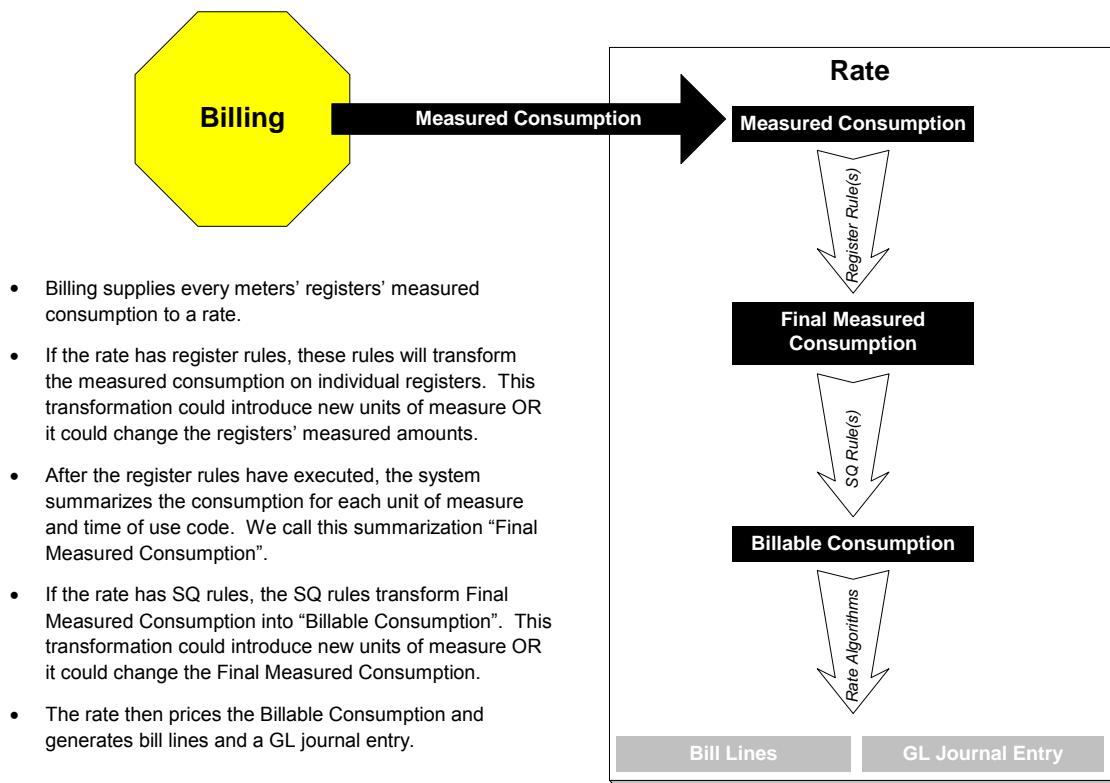
All of these examples share the same trait: the unit of measure being priced is not the same as the unit of measure being measured. If any of your rates possess this trait, then you have to tell the system about it so that it knows to perform extra manipulations before it calculates charges.

So, how do you tell the system to manipulate measured consumption? You have two options:

- Create a Service Quantity (SQ) Rule and link it to the rate schedule
- Create a Register Rule and link it to the rate schedule

When a rate schedule has either type of rule, the system executes the rule's algorithm immediately before it calculates charges. The rule's algorithms manipulate measured consumption so that the rate is supplied with consumption in the manner that it expects.

You must understand the difference between these two types of rules before you can set them up. The following diagram describes how the two types of rules are used and this should help you understand the differences between them:



**Bottom line.** Register rules are used when your rates require the manipulation of measured consumption and this manipulation requires attributes directly related to either the service point, the meter or the premise. SQ rules also manipulate measured consumption. However, this manipulation is done without regard to the individual service points or meters.

The following table defines which type of rule would be used for each of the earlier manipulation examples:

Situation	Type of Rule	Explanation
Cubic Feet (CCF) to Therm conversion	Register rule	A register rule is used because a given bill could contain many registers that measure CCF and each register could have a different therm conversion factor. If an SQ rule was used, a single conversion factor would be applied to the total CCF from all registers.
Minimum contract quantity	SQ rule	An SQ rule is used because we want to add together the total amount of consumption from all registers and then compare it to a contract quantity.
Add together peak and off peak consumption	SQ rule	An SQ rule is used because we want to add together the total amount of consumption from all registers.

## Setting Up Register Rules

**Warning!** Do NOT attempt to set up register rules until you have a very good understanding of what your meters measure and how your rates charge for this consumption. In addition, you should completely understand the concepts described in [When And How To Manipulate Measured Consumption](#) before creating these rules.

To set up register rules, choose **Admin Menu, Register Rule**.

### Description of Page

Enter a unique **Register Rule ID** and a **Description** for the register rule.

Enter **Comments** that describe, in detail, what the rule does.

Select the **Algorithm** that controls how the system manipulates the consumption supplied to the rate. Define the **Parameters** to be used by the register algorithm.

The following table indicates the algorithms that are supplied with the system, its Algorithm Type and how each register algorithm uses its parameters.

### Contents

[Base Package Register Rules](#)

[How To Setup A Register Rule To Adjust A Cubic Foot Of Gas Based On Pressure](#)

[How To Set Up A Reg. Rule To Convert Cubic Feet To Therms](#)

[How To Set Up A Register Rule To Price Interval Consumption Using Interval Prices](#)

### Base Package Register Rules

**Note.** If the register rule algorithms that are supplied with the base package are not sufficient for your rating requirements, you will have to write a new register rule algorithm. Refer to [How To Add A New Register Rule](#) for how to do this.

Register Algorithm	Alg Type	What It Does	Parameters Used
<i>Bill Factor Based UOM Conversion</i>	<a href="#">BU</a>	<p>Converts a measured quantity into a final quantity using the value specified on a bill factor.</p> <p>Final Quantity (expressed in the UOM in param 2) =</p> $\text{Quantity for UOM (in param 1)} * \text{Value for Bill Factor (in param 3)}$ <p>This algorithm would be useful when you need to convert cubic feet (CCF) into therms based on the takepoint from which the premise gets its gas.</p>	Param 1 = Measured UOM Param 2 = Final UOM Param 3 = Bill Factor containing conversion factor
<i>Real Time Pricing</i>	<a href="#">RT</a>	<p>Converts a measured quantity (UOM) into a final quantity (Result SQL) by multiplying the measured quantity by the price (Bill Factor) effective on the read's start date/time.</p> <p>Final Quantity (expressed in the SQL in param 3) =</p> $\text{Quantity for UOM (in param 1)} * \text{Value for Real Time Bill Factor (in param 2)}$	Param 1 = UOM Param 2 = Bill Factor Param 3 = Resultant SQL

		Note, this option is useful if you have simple real time pricing algorithms and do not use the interval billing subsystem. Refer to <a href="#">Pricing Interval Consumption Using Interval Prices</a> for more information.	
<i>Seasonal Register TOU Conversion</i>	<a href="#"><u>ST</u></a>	<p>Converts time-of-use codes of "current season" and "prior season" to "winter" and "summer" based on the billing period.</p> <p><b>Summer Begin Date</b> contains the MMDD when summer starts.</p> <p><b>Winter Begin Date</b> contains the MMDD when winter starts.</p> <p><b>Current Season TOU</b> contains the TOU code of the register that measures the current season's consumption. <b>Prior Season TOU</b> contains the TOU code of the register that measures the prior season's consumption. <b>Winter TOU</b> contains the TOU code to be used for winter consumption (the register read's TOU code is changed to this for winter consumption). <b>Summer TOU</b> contains the TOU code to be used for summer consumption (the register read's TOU code is changed to this for summer consumption).</p>	Param 1 = Summer Begin Date (MMDD) Param 2 = Winter Begin Date (MMDD) Param 3 = Current Season TOU Param 4 = Prior Season TOU Param 5 = Winter TOU Param 6 = Summer TOU

For more information about defining algorithm types, refer to [Setting Up Algorithm Types](#).

### Where Used

A primary Rate Schedule may have zero or more register rules. Refer to [Rate Schedule - Register Rules](#) for more information.

**How To Setup A Register Rule To Adjust A Cubic Foot Of Gas Based On Pressure**  
Refer to [Gas Pressure Factor Adjustments](#) for an example.

### How To Set Up A Reg. Rule To Convert Cubic Feet To Therms

Let's assume a gas rate contains therm-oriented charges, but the customer's meters only measure in cubic feet. This means that you will need a register rule to convert the measured consumption into therm-oriented consumption.

The following table shows the array that will be supplied to the rate assuming the customer's meter measures CCF:

Unit Of Measure	Time Of Use	Measured Quantity	Final Quantity
CCF		1000	1000

Now, let's assume the rate contains charges based on therms. This means that the rate needs to have the array changed as follow:

Unit Of Measure	Time Of Use	Measured Quantity	Final Quantity
CCF		1000	1000
TH			1010

As you can see, a new entry was added to the array. This new entry contains the amount of therms associated with the measured CCF. To do this you would need to create a register rule with the following attributes (and link it to the appropriate rates):

Field	Value
Register Algorithm	<i>Bill factor based UOM conversion</i>
UOM (Source)	<i>CCF</i>
UOM (Result)	<i>TH</i>
Bill Factor	<i>CCFCONV</i>

In addition, you will also have to set up the following bill factor and characteristic information:

- Add a characteristic type to define your various take points (call it, TAKEPT). Indicate this characteristic is specified on premises.
- Define characteristic values for the take point characteristic. These will be the take points in your service area.
- Add the bill factor you identified in the register rule (i.e., CCFCONV). Indicate that the type of characteristic to use is the take point characteristic type (i.e., TAKEPT). Add bill factor characteristics for each take point.
- Add a bill factor value for each bill factor characteristic. This value will contain the CCF to therm conversion factor to be used for the specific take point.
- Update the relevant postal defaults so that the appropriate take point will default onto your premises.
- Update existing premises with their respective take point (by specifying the appropriate characteristic value for the take point characteristic).

For more information about bill factors and characteristics, refer to [Setting Up Bill Factors](#) and [Setting Up Characteristic Types & Their Values](#).

## How To Set Up A Register Rule To Price Interval Consumption Using Interval Prices

Refer to [Pricing Interval Consumption Using Interval Prices](#) for an example.

## Setting Up Service Quantity (SQ) Rules

**Warning!** Do NOT attempt to set up SQ rules until you have a very good understanding of what your meters measure and how your rates charge for this consumption. In addition, you should completely understand the concepts described in [UOM versus TOU versus SQI](#) before creating these rules. In addition, you should completely understand the concepts described in [When And How To Manipulate Measured Consumption](#) before creating these rules.

To set up Service Quantity (SQ) rules, choose **Admin Menu, Service Quantity Rule**.

### Description of Page

Enter a unique **SQ Rule ID** and a **Description** for the SQ rule.

Enter a **Long Description** that describes, in detail, what the rule does.

Indicate the **UOM**, **TOU** and **SQI** to be output by the rule.

Every service quantity supplied to a rate has two quantities – the amount of service as it was initially measured, and the amount of service that should be priced by the rate. Most of the time, these two amounts will be the same. They will only differ when another SQ Rule has executed and caused the billable quantity to change (an SQ Rule NEVER changes initially measured quantities). Because these two amounts may differ, an SQ Rule needs to know if it should use the quantity as initially measured OR if it should use the quantity that will be priced in the rate. You use the **SQ Quantity to Use** field to indicate such. Valid values are **Use Initial Quantity** and **Use Billable Quantity**.

Indicate whether the system should **Create Billing Error** if the SQ Rule is not able to execute successfully. For most SQ Rules you would turn this option on. However, if the rule is linked to a rate where the rule is only applicable to a subset of the customer base using the rate, you may not want a billing error to be issued if the rule cannot execute. For example, consider a rule used to derive total kWh by adding together peak kWh and off-peak kWh. You would only use such a rule if you measured peak and off-peak consumption, but you had prices in respect of total consumption. In the situation where this SQ Rule was linked to a rate used by customers with and without time-of-use meters, you would not want the system to issue a billing error for those customers without time-of-use meters (because their meter already measures total consumption).

Use **SQ Rule Processing** to indicate if the SQ rule is **Always** executed by rate application, even if the consumption details on the original bill segment are to be used when the system recalculates a bill segment (refer to [Rebill \(Bill Segment\)](#)). If the SQ rule should not be executed when the consumption details on the original bill segment are used, indicate that the SQ rule should only be executed on the **Initial SQ Calculation**. SQ rules that create a characteristic type / value (in the characteristic collection) used by one or more bill factors referenced in a rate component should typically be always executed. Refer to [Base Package SQ Rules](#) for examples of SQ rules that create a characteristic type / value.

Select the **SQ Algorithm Type** that controls how the system manipulates the consumption supplied to the rate. Define the **Parameters** to be used by the SQ algorithm.

## Contents

[Base Package SQ Rules](#)

[How To Set Up An SQ Rule To Add Together Two Service Quantities](#)

### Base Package SQ Rules

The following table describes the SQ rule algorithms that are supplied with the system.

**Note.** If the algorithms described below are not sufficient for your rating requirements, you must write a new SQ rule algorithm. Refer to [How To Add A New SQ Rule](#) for how to do this.

SQ Algorithm	Alg Type	What It Does	Parameters Used
<a href="#">Bill Credits for Electric Consolidated Billing</a>	<a href="#">BC</a>	<p>This SQ rule creates a Characteristic Type / Value to handle consolidated billing credits. These values are used by one or more bill factors referenced in the rate components to vary the amount associated with a given bill line.</p> <p>The characteristic type and value are generated if a service provider (SPr) provides consolidated billing for its customer. These credits are applied irrespective of whether the SPr</p>	Param 1 = SA Relationship Type Param 2 = Service Type for Electric Param 3 = Characteristic Type - Billing Credit Param 4 = Characteristic Value - BR Dual Commodity Value

SQ Algorithm	Alg Type	What It Does	Parameters Used
		<p>provides full or partial consolidated billing and whether or not the customer takes both gas and electric services.</p> <p><b>SA Relationship Type</b> tells the system the type of SA relationship (e.g., energy service provider versus meter agent) that is interrogated.</p> <p><b>Service Type</b> tells the system the service type associated with electricity (as the characteristic value returned differs for electric-only service providers).</p> <p>The Characteristic collection will be updated with the <b>Characteristic Type</b> and an appropriate Value as follows:</p> <ul style="list-style-type: none"> <li>• <b>BR Dual Commodity Value</b> if the SPr provides bill ready consolidation for more than one service</li> <li>• <b>BR Electric Only Value</b> if the SPr provides bill ready consolidation for electric only</li> <li>• <b>RR Dual Commodity Value</b> if the SPr provides rate ready consolidation for more than one service</li> <li>• <b>RR Electric Only Value</b> if the SPr provides rate ready consolidation for electric only</li> </ul>	Param 5 = Characteristic Value - BR Electric Only Value Param 6 = Characteristic Value = RR Dual Commodity Value Param 7 = Characteristic Value = RR Electric Only Value
<i>Bill Credits for Gas Consolidated Billing</i>	<a href="#">CT</a>	<p>This SQ rule creates a characteristic type / value if a service provider provides consolidated billing for the customer being billed. These values are used by one or more bill factors in the rate components to vary the amount associated with a given bill line.</p> <p>The characteristic type and value are generated if a customer's gas service provider (SPr) provides consolidated billing (i.e., the service provider has a billing relationship of "<i>they bill for us</i>").</p> <p>The <b>Service Type</b> tells the system how to identify gas service.</p> <p>The <b>SA Relationship Type</b> tells the system the type of SA relationship (e.g., energy service provider versus meter agent) that is interrogated.</p> <p>If the above conditions are true, the Characteristic collection will be updated with the <b>Characteristic Type</b> and an appropriate Value as follows:</p> <ul style="list-style-type: none"> <li>- <b>CTA Credit Residential</b> if the customer class matches the <b>Customer Class for Residential</b></li> <li>- <b>CTA Credit Non-Residential</b> if the customer class does not match the <b>Customer Class for Residential</b></li> </ul>	Param 1 = Service Type Param 2 = SA Relationship Type Param 3 = Customer Class for Residential Param 4 = Characteristic Type - CTA Credit Param 5 = Characteristic Value - CTA Credit Residential Param 6 = Characteristic Value - CTA Credit Non-Residential
<i>Bill days (Days)</i>	<a href="#">DY</a>	This SQ rule calculates the total number of days in the bill period. This algorithm is necessary if you have charges that are based on the number of days of service (e.g., a minimum charge that varies based on the days of service). Refer to <a href="#">Calculated Minimum Charges</a> and <a href="#">Step Boundaries Based On The Number Of Days Of Usage And The Region In Which The Customer</a>	No parameters are used.

SQ Algorithm	Alg Type	What It Does	Parameters Used
		<a href="#">Resides</a> for good examples of when to use such an SQ rule.	
<i>Characteristic Value</i>	<a href="#">CV</a>	<p>This SQ rule creates an entry in the SQ array containing the sum of the characteristic values of a given Characteristic Type. The values are accumulated from the entity defined in the parameter list.</p> <p>Resulting Quantity (expressed in the Output UOM / TOU / SQL) = the sum of the characteristic values of a given <b>Characteristic Type</b>. <b>Characteristic Source</b> determines the entity that holds the characteristic values:</p> <ul style="list-style-type: none"> <li>• If <b>Characteristic Source</b> is "SA", the characteristic value effective on the bill start date for the Characteristic Type (P1) and SA Id is returned.</li> <li>• If <b>Characteristic Source</b> is "ACCOUNT", the characteristic value effective on the bill start date for the Characteristic Type (P1) and Account Id is returned.</li> <li>• If <b>Characteristic Source</b> is "PREMISE", the characteristic values are accumulated from all premises linked to the SA ID.</li> <li>• If <b>Characteristic Source</b> is "SP", the characteristic values are accumulated from all service points linked to the SA ID.</li> </ul> <p>If <b>Minimum Value</b> is greater than zero and the resultant characteristic value is less than the specified <b>Minimum Value</b>, an error is generated.</p>	Param 1 = Characteristic Type Param 2 = Characteristic Source Param 3 = Minimum Value
<i>Check Load Factor</i>	<a href="#">LF</a>	<p>This SQ rule generates a To Do entry if a customer's actual demand exceeds a fixed demand amount (and actual kWh exceeds a calculated kWh).</p> <p>The actual algorithm is as follows:</p> <p>Retrieve maximum demand from SQ collection as the greatest of <b>kW/Peak</b>, <b>kW/Part Peak</b> and <b>kW/Off-Peak</b>. If this maximum demand &lt;= <b>Threshold Maximum Demand</b>, escape the routine. If this maximum demand &gt; <b>Threshold Maximum Demand</b>:</p> <ul style="list-style-type: none"> <li>• Compute 'Calculated kWh' = <b>Threshold Maximum Demand</b> * total days in bill period, times 24.</li> <li>• If <b>Actual kWh/Calculated kWh</b> &gt; 1, create a To Do Entry (using the <b>To Do Type</b> and <b>To Do Role</b> (if specified)). The base package includes the <b>To Do Type LF</b> to use for this parameter value, unless you've set up your own.</li> </ul>	Param 1 = UOM of demand (e.g., KW) Param 2 = TOU of peak period Param 3 = TOU of off-peak period Param 4 = TOU of part peak period Param 5 = UOM of actual KWH (e.g., KWH) Param 6 = Threshold maximum demand Param 7 = <b>To Do Type</b> Param 8 = <b>To Do Role</b> (optional)
<i>Consumption History</i>	<a href="#">CH</a>	<p>This SQ rule adds an entry in the SQ array that contains a given amount of historical consumption.</p> <p>This algorithm can return <b>MAX</b>(maximum), <b>MIN</b>(minimum), <b>SUM</b>(total) or <b>AVG</b> (daily average) consumption as defined by the <b>Operator</b>. It retrieves this consumption from the SA's historical bill segments (from the service quantities that are</p>	Param 1 = Season Start Month / Day (MM/DD) Param 2 = Season End Month / Day (MM/DD) Param 3 = Number of Historical Billing Periods

SQ Algorithm	Alg Type	What It Does	Parameters Used
		<p>stored on the bill segments).</p> <p><b>UOM / TOU / SQI</b> define the type of historical consumption that is retrieved.</p> <p>The historical period whose consumption is accessed is defined by EITHER:</p> <ul style="list-style-type: none"> <li>• <b>Number of Historical Periods</b> (a period is defined by the rate schedule's frequency) or</li> <li>• The previous historical season (defined by <b>Start Month/Day</b> and <b>End Month/Day</b>).</li> </ul> <p>PLEASE SPECIFY EITHER OF THE ABOVE PARAMETERS.</p> <p><b>Note.</b> If the system finds historical bills with a period outside the normal period for the rate's frequency, proration takes place if the input UOM does not Measure Peak. The algorithm accesses consumption data starting from the most recent bill going backwards. As soon as an irregular bill period is encountered, the system attempts to determine appropriate consumption for a "normal" period by prorating the irregular bill based on actual data from previous bills.</p> <p>For example, if historic bills for a monthly rate cover</p> <ul style="list-style-type: none"> <li>▪ Jan 1 - Feb 1</li> <li>▪ Feb 1 - Mar 1</li> <li>▪ Mar 1 - Mar 15</li> </ul> <p>When the next bill is generated, the system attempts to calculate the historic consumption going backwards. It will determine consumption for Feb 15 - Mar 15 by taking the consumption for the March bill and adding consumption the time from Feb 15 to Mar 1 by prorating the consumption on the February bill. Then, it attempts to determine consumption for Jan 15 - Feb 15 by taking the remaining consumption for February and adding consumption from the time from Jan 15 to Feb 1 by prorating the consumption on the January bill, etc. Once enough historical bills have been processed, any days "leftover" from the earliest bill as a result of this proration logic are simply ignored.</p> <p>If the input UOM measures peak, no proration of historical quantities occurs. If no UOM is specified as input (i.e., only an SQI), the system assumes that the quantity should behave as if it does not measure peak.</p>	Param 4 = UOM Param 5 = TOU Param 6 = SQI Param 7 = Operator
<i>Contract Quantity</i>	<a href="#">CQ</a>	<p>This SQ rule is used to create an entry in the SQ array for a given Contract Quantity Type specified in a SA. If multiple values are effective during a bill period, the Proration Rule determines what happens. (See below).</p> <p>Resulting Quantity (expressed in the Output UOM / TOU / SQI) = the value of a given <b>Contract Quantity Type</b> in a SA. If multiple values are effective during a bill period the <b>Proration Rule</b></p>	Param 1 = Contract Quantity Type Param 2 = Proration Rule: CQBD, CQED, CQMA, CQMI, CQPR

SQ Algorithm	Alg Type	What It Does	Parameters Used
		<p>determines what happens:</p> <ul style="list-style-type: none"> <li>If Proration Rule is 'Use Contract Quantity effective at Beginning of Bill Period' (i.e., <b>CQBD</b>), SQ Amount = Contract Quantity effective at the start of the bill period.</li> <li>If Proration Rule is 'Use Contract Quantity effective at End of Bill Period' (i.e., <b>CQED</b>), SQ Amount = Contract Quantity effective at the end of the bill period.</li> <li>If Proration Rule is 'Maximum' (i.e., <b>CQMA</b>), SQ Amount = largest Contract Quantity effective in the bill period.</li> <li>If Proration Rule is 'Minimum' (i.e., <b>CQMM</b>), SQ Amount = smallest Contract Quantity effective in the bill period.</li> <li>If Proration Rule is 'Prorate' (i.e., <b>CQPR</b>), SQ Amount = Sum of all prorated contract quantities (calculate the number of days in the bill period (X); calculate the number of days in the bill period that each contract quantity is effective (Y(i)); prorate each contract quantity (i) = (Contract Quantity(i) * Y(i) / X)).</li> </ul>	
<i>Direct Access Credits</i>	<b>CD</b>	<p>This SQ rule creates a characteristic type / value to handle credits issued to the customer for participating in direct access. These values are used by one or more bill factors in the rate components to vary the amount associated with a given bill line.</p> <p>This rule always creates a characteristic type equal to <b>Characteristic Type DA Credit</b>. The characteristic value will be set as follows:</p> <ul style="list-style-type: none"> <li>If the service agreement being billed has a service agreement relationship with either the <b>ESP Relationship Type</b> or the <b>CTA Relationship Type</b>, a characteristic value of <b>Characteristic Value - DA Customer</b> will be created.</li> <li>If not, a characteristic value of <b>Characteristic Value - Non DA Customer</b> will be created.</li> </ul>	Param 1 = ESP Relationship Type Param 2 = CTA Relationship Type Param 3 = Characteristic Type - DA Credit Param 4 = Characteristic Value - DA Customer Param 5 = Characteristic Value - Non DA
<i>Early Pay</i>	<b>EP</b>	<p>This SQ rule is only used if you provide customers with early payment discounts via a discount on their next bill segment. This rule works as follows:</p> <ul style="list-style-type: none"> <li>First, it determines if the prior bill segment's bill was paid before its due date. It does this by retrieving the customer's balance as of the prior bill's due date - 1 day. If this amount is &lt;= zero, it checks if a payment exists with a payment date on or after the bill's bill date and before the bill's due date. If yes, processing continues.</li> <li>A characteristic type / value is created where the value holds the number of days early (and the characteristic type is equal to <b>Char Type - Number of Days Early</b>).</li> <li>In addition, this routine accumulates the sum of the bill lines</li> </ul>	Param 1 = Char Type - Number of Days Early Param 2 = Char Type - Eligible For Early Pay Disc Param 3 = Char Value - Eligible For Early Pay Disc

SQ Algorithm	Alg Type	What It Does	Parameters Used
		<p>on the prior bill segment for which the early payment discount is eligible. It does this by accumulating all bill lines that reference a given characteristic type and value (where characteristic type is equal to Char Type - Eligible For Early Pay Disc, and Char Value - Eligible For Early Pay Disc). It populates the total amount in the service quantity defined on the SQ rule. This means that you must define this characteristic type / value on each rate component whose resultant value is eligible to receive an early payment discount.</p> <p>This rule anticipates the early pay discount being levied using two rate components:</p> <ul style="list-style-type: none"> <li>The first would be an FCPO rate component that holds the amount eligible to receive the discount. This would be an SQ rate component where the service quantity is the value produced by this SQ rule, and the price would be 1 (or -1). In addition, this rate component would need eligibility rules so that it is only applied if the number of days early (held in the characteristic type / value produced by this algorithm) is &gt; 0.</li> <li>The second would be an "apply to" rate component (where the apply to value is a percentage applied to the first rate component). In addition, this rate component would need eligibility rules so that it is only applied if the number of days early is within a certain tolerance (e.g., &gt; 0). You could create several of these types of rate components if the discount percent differs based on the number of days early.</li> </ul>	
<i>Mathematical Functions</i>	<u>MA</u>	<p>This SQ rule performs mathematical functions on service quantities prior to rate application.</p> <p>There are three major groups of parameters:</p> <ol style="list-style-type: none"> <li>1) <b>Quantity 1</b> (identified by a service quantity's UOM/TOU/SQI).</li> <li>2) <b>Mathematical operand</b></li> <li>3) <b>Quantity 2</b> (identified by a service quantity's UOM/TOU/SQI or a Bill Factor)</li> </ol> <p>For example, if you want to add together two SQ entries, these values would contain:</p> <ol style="list-style-type: none"> <li>1) the unique identifier of one of the SQ's (in parameters 1 through 3)</li> <li>2) the "+" symbol (in parameter 4)</li> <li>3) the unique identifier of the other SQ (in parameters 5 through 7)</li> </ol> <p>It should be stressed that <b>Quantity 2</b> can either reference a <b>bill factor</b> (param 8) or an SQ entry (parameters 5 through 7). If both are specified, the bill factor is used.</p>	<p>Quantity 1:</p> <p>Param 1 = UOM of service quantity entry</p> <p>Param 2 = TOU of service quantity entry</p> <p>Param 3 = SQI of service quantity entry</p> <p>Or</p> <p>Param 9 = Decimal number</p> <p>Mathematical Operand:</p> <p>Param 4 = operand symbol (i.e., +, -, *, /)</p> <p>Quantity 2:</p> <p>Param 5 = UOM of service quantity entry</p> <p>Param 6 = TOU of service</p>

SQ Algorithm	Alg Type	What It Does	Parameters Used
		<p>The two <b>default values</b> (param 9 and param 10) are only used if Quantity 1 and/or Quantity 2 reference SQ's / bill factor that are not present at billing time. There are three ways to use these <b>default values</b>:</p> <ul style="list-style-type: none"> <li>- If you want to multiply a SQ by a constant value, define the SQ in Quantity 1 and the constant value in Param 10, i.e., you don't have to specify anything in Param's 5 through 8.</li> <li>- If you want a default value used if the system cannot find a SQ that matches the UOM/TOU/SQI, define this value in Param 9 and/or Param 10.</li> <li>- If you use a bill factor to define Quantity 2 (in Param 8) and you want a default value used if the system cannot find a char type / value for a specific customer, the default value is defined in Param 10.</li> </ul> <p>The following <b>mathematical operands</b> use only <b>Quantity 1</b>.</p> <p>ABS, NEGATE, ROUND, FLOOR (meaning to round down), CEILING (meaning to round up), SIN, ASIN, COS, ACOS, TAN, ATAN, LOG, LOG10, EXP, EXP10, SQRT</p> <p>The following <b>mathematical operands</b> use both <b>Quantity 1</b> and <b>Quantity 2</b>:</p> <p>+, -, *, /, **, MOD, &lt;, &lt;=, &gt;, &gt;=, =, &lt;&gt;, MAX, MIN, AVG</p> <p>The following <b>mathematical operands</b> use both <b>Quantity 1</b> and <b>Quantity 2</b> and return a value of 1 if the comparison is true and a value of 0 if the comparison is false:</p> <p>&lt;, &lt;=, &gt;, &gt;=, =, &lt;&gt;</p>	quantity entry Param 7 = SQL of service quantity entry Or Param 8 = Bill factor ID Or Param 10 = Decimal number
<i>Maximum Service Quantity</i>	<u>MQ</u>	<p>This SQ rule compares a quantity (either measured or created via a previous SQ rule) against a contract quantity and returns whichever is larger.</p> <p>Resulting Quantity (expressed in the Output UOM / TOU / SQI) = greater of quantity for <b>UOM / TOU / SQI</b>  OR  service agreement quantity for <b>Contract Quantity Type</b></p>	Param 1 = UOM Param 2 = TOU Param 3 = SQI Param 4 = Contract Quantity Type
<i>Meter Ownership</i>	<u>MO</u>	<p>This SQ rule returns SQI = number of meters that are not owned by the company per day.</p> <p>It applies if a service provider owns a direct access customer's meter or a customer owns its own meter.</p> <p>The system expects an SA Relationship to exist with the <b>Meter Ownership/Services Relationship Type</b>. If a Relationship exists where the service provider is a third party, the number of meters linked to this SA is counted and returned.</p>	Param 1 = Meter Ownership/Services Relationship Type
<i>Meter Service Credits</i>	<u>MD</u>	<p>This SQ rule creates a characteristic type / value to handle credits issued to the customer because a customer's meter is</p>	Param 1 = Meter Reading Relationship Type

SQ Algorithm	Alg Type	What It Does	Parameters Used
		<p>read by a 3<sup>rd</sup> party service provider. These values are used by one or more bill factors in the rate components to vary the amount associated with a given bill line.</p> <p>It applies if a service provider (SPr) provides meter reading services for its customer. These credits are applied irrespective of whether or not the customer takes both gas and electric service and the method the SPr uses to read the meter.</p> <p><b>Meter Reading Relationship Type</b> tells the system the type of SA relationship (e.g., energy service provider versus meter agent) that is interrogated.</p> <p><b>Service Type</b> tells the system the service type associated with electricity (as the characteristic value returned differs for electric-only service providers).</p> <p>The Characteristic collection will be updated with the <b>Characteristic Type</b> and an appropriate Value as follows:</p> <ul style="list-style-type: none"> <li>• <b>MDMA Dual Commodity Value</b> if the ESP provides meter reading for more than one service</li> <li>• <b>MDMA Electric Only Value</b> if the ESP provides bill ready consolidation for electric only</li> </ul>	Param 2 = Service Type for Electric Param 3 = Characteristic Type - MDMA Credit Param 4 = Characteristic Value - MDMA Dual Commodity Value Param 5 = Characteristic Value - MDMA Electric Only Value
<i>Number of Meters</i>	<u>NM</u>	This SQ rule calculates the number of meters (actually service points) linked to the service agreement.	No parameters are used
<i>Power Factor</i>	<u>PF</u>	<p>This SQ rule calculates power factor. It will also create a <b>To Do</b> entry if the resultant power factor exceeds a user-defined threshold value.</p> <p>Resulting Quantity (expressed in the Output UOM / TOU / SQL) = Average Power Factor Percentage (N3 format, no decimals) = COSINE (ARCTANGENT (value of UOM4/TOU5/SQI6 / value of UOM1/TOU2/SQI3)) * 100</p> <p>If the calculated power factor exceeds <b>Power Factor Threshold Percent</b>, a <b>To Do</b> entry is created (using the <b>To Do Type</b> and <b>To Do Role</b>). The base package includes the <b>To Do Type PF</b> to use for this parameter value, unless you've set up your own.</p>	Service Quantity ID of kWh <ul style="list-style-type: none"> <li>• Param 1 = UOM1</li> <li>• Param 2 = TOU1</li> <li>• Param 3 = SQI1</li> </ul> Service Quantity ID of kVARh <ul style="list-style-type: none"> <li>• Param 4 = UOM2</li> <li>• Param 5 = TOU2</li> <li>• Param 6 = SQI2</li> </ul> Param 7 = <b>To Do Type</b> Param 8 = <b>To Do Role</b> Param 9 = Power Factor Threshold Percent
<i>Required UOM/TOU Validation</i>	<u>RP</u>	<p>This SQ rule verifies if one UOM/TOU quantity is non-zero then the other should be non-zero too.</p> <p><b>UOM1/TOU1/SQI1</b> is the ID of the first SQ entry.  <b>UOM2/TOU2/SQI2</b> is the ID of the second SQ entry.</p>	Service Quantity ID1 <ul style="list-style-type: none"> <li>• Param 1 = UOM1</li> <li>• Param 2 = TOU1</li> <li>• Param 3 = SQI1</li> </ul> Service Quantity ID2 <ul style="list-style-type: none"> <li>• Param 4 = UOM2</li> </ul>

SQ Algorithm	Alg Type	What It Does	Parameters Used
			<ul style="list-style-type: none"> <li>Param 5 = TOU2</li> <li>Param 6 = SQI2</li> </ul>
<i>Season Days</i>	<u>SD</u>	<p>This SQ rule is used to create an entry in the SQ array that contains the number of days in the current bill period that fall in a defined season.</p> <p>Resulting Quantity (expressed in the Output UOM / TOU / SQI) = number of days in the current bill period that fall in <b>Season Start</b> and <b>Season End</b>. For example, if bill period = 03/15/99 to 04/15/99 and Season Start date = 04/01 and Season End date = 10/31, then resulting quantity = 15.</p>	Param 1 = Season Start (MMDD) Param 2 = Season End (MMDD)
<i>Seasonal Usage</i>	<u>SU</u>	<p>This SQ rule is responsible for taking a non-seasonal UOM/TOU/SQI and producing a seasonal UOM/TOU/SQI, depending on the bill period. For example, if you have a meter which only measures on and off peak consumption, but a rate that bills for summer on peak and summer off peak consumption, then this rule will be used.</p> <p>Resulting value = value of UOM1/TOU1/SQI1 * (value of UOM2/TOU2/SQI2 / value of UOM3/TOU3/SQI3)</p>	Normal consumption Param 1 = UOM1 Param 2 = TOU1 Param 3 = SQI1 Number of days in the season Param 4 = UOM2 Param 5 = TOU2 Param 6 = SQI2 Number of days in the billing period Param 7 = UOM3 Param 8 = TOU3 Param 9 = SQI3
<i>Sum</i>	<u>SM</u>	<p>This SQ rule adds up several quantities (either measured or created via a previous SQ rule). This algorithm would be useful when you need to add together "time-of-use" registers because there are charges in the rate that are for the total amount of consumption regardless of when it was used.</p> <p>Resulting Quantity (expressed in the Output UOM / TOU / SQI) =</p> <p style="padding-left: 40px;">Quantity for UOM (in param 1) TOU (in param 2) SQI (in param 3) +</p> <p style="padding-left: 40px;">Quantity for UOM (in param 4) TOU (in param 5) SQI (in param 6) +</p> <p style="padding-left: 40px;">Quantity for UOM (in param 7) TOU (in param 8) SQI (in param 9)</p>	Quantity 1 Param 1 = First UOM Param 2 = First TOU Param 3 = First SQI Quantity 2 Param 4 = Second UOM Param 5 = Second TOU Param 6 = Second SQI Quantity 3 Param 7 = Third UOM Param 8 = Third TOU Param 9 = Third SQI

For more information about defining algorithm types, refer to [Setting Up Algorithm Types](#).

## Where Used

A Rate Schedule may have zero or more SQ Rules. Refer to [Rate Schedule - SQ Rules](#) for more information.

### How To Set Up An SQ Rule To Add Together Two Service Quantities

Let's assume the rate contains charges based on the TOTAL amount of kilowatt hours (i.e., there is a price expressed per kWh with no TOU or SQI). However, some customers who use this rate have time-of-use meters that measure kWh in respect of peak and off-peak hours (i.e., the meters have two registers where one measures kWh peak and the other measures kWh off-peak). This means that you will need an SQ Rule to add together the two measured service quantities to derive total kWh.

The following table shows the array that will be supplied to the rate assuming the customer's meter measures kWh peak usage and kWh off peak usage:

Unit Of Measure	Time Of Use	SQ Identifier	Initial Quantity	Billable Quantity
KWH	PEAK		1000	1000
KWH	OFFPEAK		800	800

Now, let's assume the rate contains charges based on the TOTAL amount of kilowatt hours (i.e., the sum of KWH PEAK and KWH OFFPEAK). This means that the rate needs to have the array changed as follow:

Unit Of Measure	Time Of Use	SQ Identifier	Initial Quantity	Billable Quantity
KWH	PEAK		1000	1000
KWH	OFFPEAK		800	800
KWH				1800

As you can see, a new entry was added to the array. This new entry contains the sum of two existing service quantities. Notice that the new entry doesn't have an initial quantity. This is because an SQ Rule will create it and SQ Rules never populate initial quantity (only metered consumption can exist in initial quantity). To add this new entry, create an SQ Rule with the following attributes (and link it to the appropriate rates):

Field	Value
SQ Algorithm	<i>Sum</i>
UOM 1	<i>KWH</i>
TOU 1	<i>PEAK</i>
SQI 1	
UOM 2	<i>KWH</i>
TOU 2	<i>OFFPEAK</i>
SQI 2	
UOM 3	
TOU 3	
SQI 3	

Quantity to Use	<b>Billable Quantity</b> (actually, it doesn't matter because both the Billable and Initial Quantities will contain the same value in this scenario).
Create a Billing Error	It depends on your situation. If all customers using this rate have time-of-use meters that need to be summed together to come up with the total kWh, then you'd want to create a billing error if the system is not supplied with the appropriate service quantities. If some customers using this rate have time-of-use meters and others just have summary meters that don't differentiate based on time-of-use, you would NOT select this option because you wouldn't want a billing error generated for customers without a time-of-use meter.

## Defining Service Quantity (SQ) Identifiers

Consider a rate that contains a charge per "meter day" where the total number of meter days equals the number of days in the billing period multiplied by the number of customer-owned meters at the premise. At billing time, you'd need an SQ Rule to count the number of customer-owned meters and multiply this value by the number of days in the billing period. You could instruct the SQ Rule to place the resultant number of meter days in some unit of measure (meaning you'd have to create a UOM called "meter day"). Alternatively, you could instruct the SQ Rule to place the resultant number in a service quantity identifier (SQI) (meaning you'd have to create an SQI called "meter day"). The choice is yours. As a rule of thumb, we'd recommend creating an SQI for those things that are not measured by a meter.

**Warning!** SQIs exist to support calculations performed by rate components. This means that you must design your rate components before you can design your service quantity identifiers. We strongly recommend that you design "on paper" how every rate's rate component looks before you attempt to set up your SQIs.

**Note.** If you create an SQI, you must also have a corresponding SQ Rule to calculate the amount of the SQI to be priced by the rate. If not, it is impossible for the system to generate a bill line for the SQI because there will be no quantity associated with the SQI.

To define service quantity identifiers, open **Admin Menu, Service Quantity Identifier**.

### Description of Page

Enter a unique **Service Qty. Identifier** and **Description** for every SQI.

Enter the appropriate number of **Decimal Positions** for each SQI. The rate application process uses this information to round the calculated service quantity prior to applying appropriate rate components.

### Where Used

An SQ Rule may use an SQ Identifier to describe what it generates. Refer to [Setting Up Service Quantity \(SQ\) Rules](#) for more information.

A Service Quantity Rate Component may use an SQ Identifier to describe what it is pricing. In this situation, the rate must contain an SQ Rule that generates a quantity for this SQI; otherwise there will be a price without a service quantity. Refer to [Rate Component – Main Information](#) for more information.

## Setting Up A Rate Schedule

A primary rate schedule is setup for every rate that you offer to your customers. A subsidiary rate schedule is setup to encapsulate common rate components shared by multiple primary rate schedules. Rate schedules are defined using the pages described in this section.

**A rate schedule isn't complete without rate versions and rate components.** Refer to [All Rates Share A Common Structure](#) for information about how you must also create rate versions and rate components for every rate schedule.

### Contents

- [Rate Schedule - Main](#)
- [Rate Schedule - SQ Rules](#)
- [Rate Schedule - Register Rules](#)
- [Rate Schedule - Bill Messages](#)
- [Rate Schedule - Meter Configuration Type](#)
- [Rate Schedule - Sub Rates](#)

## Rate Schedule - Main

You start the rate definition process by selecting **Main Menu, Rates, Rate Schedule**. On this page you enter general information about the rate.

### Description of Page

Enter a unique **Rate Schedule ID** to identify the rate. The value that you supply will typically correspond with the identity of the rate in your rate book.

Enter a **Description** for the rate schedule.

Select the **Rate Schedule Type** that applies to the rate schedule. Rate schedules can be either **Primary** or **Subsidiary**. Primary rate schedules are the main type of rate schedule used to calculate charges and bills for customers. Subsidiary rate schedules can be used to define rate components that are shared by more than one primary rate. For example, if an organization has multiple rate schedules that reference the same tax rate components, these tax rate components could be placed into a subsidiary rate which the primary rates would reference. Note that subsidiary rate schedules may not be referenced on service agreements, or defined as default rate schedules on SA Types. See [Use Sub Rate Schedules to Encapsulate Common Rate Components](#) for more information about subsidiary rate schedules.

**Note.** **Service Type, Frequency, Currency Code** and **Allow Estimates** defined for the primary rate schedule are used to drive the calculations at billing time and may not be specified on subsidiary rates

Select the **Service Type** that describes the service associated with this rate. When you maintain an existing rate schedule, this field will be gray if the rate is referenced on at least one SA Type. See [Setting Up Service Types](#) for more information.

Select a **Frequency** to define the time period in which the rate's charges are expressed. See [Defining Frequency Codes](#) for more information.

**Frequency may also affect the read used to bill a customer.** If you use the base package's [get consumption algorithm](#) to amalgamate reads during billing, the rates frequency also affects whether a bill segment is created AND the period of time during which the system will look for an "end read". Refer to [Meter Read Snapshot](#) for more information.

Select a **Currency Code** to define the currency in which the rate's prices are expressed. See [Defining Currency Codes](#) for more information.

**Currency note.** All bill factors that are referenced on the rate's rate components must be denominated in the same currency as defined on the rate schedule. This is because bill factors are used when the price is "soft" and all prices in a rate, be they hard or soft, must be in the same currency. Also note – all rates referenced on an account's service agreements must be denominated in the same currency as defined on the account.

Turn on **Allow Estimates** if the bill process is allowed to estimate consumption for service agreements that use this rate. Consumption is estimated if all of the following points are true:

- A real read cannot be found.
  - The service agreement allows estimation.
  - The bill cycle schedule allows estimation (if the bill is produced by the BILLING background process).
  - The rate schedule allows estimation.

Turn on **Allow RVs Proration** if a change in the Rate Version during a bill period should be prorated. If changes should not be prorated, indicate the **RV Selection Date** to use when the system determines which rate version to use; available options are:

<b><i>Accounting Date</i></b>	The rate version effective on the bill's accounting date will be used.
<b><i>Bill End Date</i></b>	The rate version effective on the bill's end date will be used.
<b><i>Bill Start Date</i></b>	The rate version effective on the bill's start date will be used.

## Rate Schedule Tree

The right half of this page is dedicated to a tree that shows the rate schedule's SQ Rules, Register Rules, Rate Versions, and Primary/Sub Rates (primary rate schedules list any linked sub rates including the sub rates SQ Rules, Rate Versions, etc.; subsidiary rate schedules list any linked primary rates). You can use this tree to view high-level information about these objects. You may also transfer to a given rate version by selecting that rate version.

**Note.** The duplicate action in the action button bar enables you to copy another rate schedule and optionally, its rate versions. Refer to [Duplicate Button](#) in the system wide standards document for more information.

### Where Used

Most SA Types contain a collection of valid Rate Schedules. Refer to [SA Type – Rate](#) for more information.

Most Service Agreements reference a Rate Schedule. Refer to [Service Agreement – Rate Info](#) for more information.

## Rate Schedule - SQ Rules

When a rate contains prices for things that aren't measured directly by the customers' meters, you must link SQ Rules to the rate. When a rate schedule has SQ Rules, the system executes each rule's algorithm before it calculates the charges embodied in the rate's rate components. These SQ Rules manipulate measured consumption so that the rate is supplied with consumption in the manner that it expects. Open **Rates, Rate Schedule** and navigate to the **SQ Rule** to link SQ Rules to a rate.

For more information, refer to [Setting Up Service Quantity \(SQ\) Rules](#).

**Note.** The [Rate Schedule Merge](#) page enables you to build the SQ rule collection for your rate schedule by copying collections from existing rate schedules.

### Description of Page

Enter a **Service Quantity Rule** for every type of service quantity manipulation required by the rate. The **Sequence** number is important when you have multiple rules linked to the rate, as it defines the order in which the rules will be executed.

### Where Used

Billing uses this information to create additional service quantities at billing time.

## Rate Schedule - Register Rules

When a meter measures something in a unit of measure that has to be converted into a different unit of measure before it can be rated, you must link register rules to the rate. A good example is when a meter measures gas in cubic feet, but the rate contains prices in therms.

When a rate schedule has register rules, the system executes each rule's algorithm before it calculates charges. These register rules convert measured consumption so that the rate is supplied with consumption in the manner that it expects.

For more information about register rules, refer to [Setting Up Register Rules](#).

**Note.** The [Rate Schedule Merge](#) page enables you to build the register rule collection for your rate schedule by copying collections from existing rate schedules.

Select **Main Menu, Rates, Rate Schedule** and navigate to the **Register Rule** to link register rules to a rate.

### Description of Page

Enter a **Register Rule** for every type of register rule to be executed before consumption is rated. The **Sequence number** is important when you have multiple rules linked to the rate, as it defines the order in which the rules will be executed.

**Note.** **Register Rules** are not allowed on subsidiary rate schedules.

#### Where Used

Billing uses this information to create or adjust service quantities at billing time.

## Rate Schedule - Bill Messages

When a rate schedule has bill messages, the system will sweep these messages onto bills that use the rate.

For more information about bill messages, refer to [The Source Of Bill Messages](#).

**Note.** The [Rate Schedule Merge](#) page enables you to build the bill message collection for your rate schedule by copying collections from existing rate schedules.

Select **Main Menu, Rates, Rate Schedule** and navigate to the **Bill Messages** to link bill messages to a rate.

#### Description of Page

Use the **Bill Messages** collection to define **Bill Message** codes that should appear on bills that use a given rate schedule. 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.

## Rate Schedule - Meter Configuration Type

**Note.** The [Rate Schedule Merge](#) page enables you to build the meter configurations collection for your rate schedule by copying collections from existing rate schedules.

Select **Main Menu, Rates, Rate Schedule** and navigate to **Meter Configuration Type** to define the types of meters that may be linked to service agreements that use this rate.

#### Description of Page

Use the **Meter Configuration Type** collection to define the recommended meter configuration types for meters linked to service agreements (by way of their service points) for a given rate schedule.

**Note.** **Meter Configuration Types** are not allowed on subsidiary rate schedules.

### Where Used

This information is used by the system to verify that meter configuration types for any meters linked to the SA's service points are valid for the SA's rate. This check exists when you change the rate for a service agreement or when you change the meter configuration type of a meter configuration that is currently linked to an active SA (via the meter and the SP). Note that if a meter configuration for the SA's service point's meter does not match one of the types in the list, a warning is issued, not an error.

## Rate Schedule - Sub Rates

Select **Main Menu, Rates, Rate Schedule** and navigate to **Sub Rates** to define any subsidiary rate schedules associated with the current (primary) rate schedule.

**Note.** Sub Rates are only allowed on Primary rate schedules. See [Use Sub Rate Schedules to Encapsulate Common Rate Components](#) for more information about subsidiary rate schedules.

### Description of Page

Use the **Sub Rate Schedule** collection to define the subsidiary rate schedules linked to the current rate schedule.

The **Sequence** is important when you have multiple sub rates linked to the primary rate, as it defines the order in which the sub rates will be calculated.

### Where Used

At billing time, if there are sub rate schedules linked to the service agreement's primary rate schedule, rate components defined in the sub rate schedules are used to calculate charges. Bill segments contain separate bill calculation details for each primary rate schedule and sub rate schedule used in its calculation.

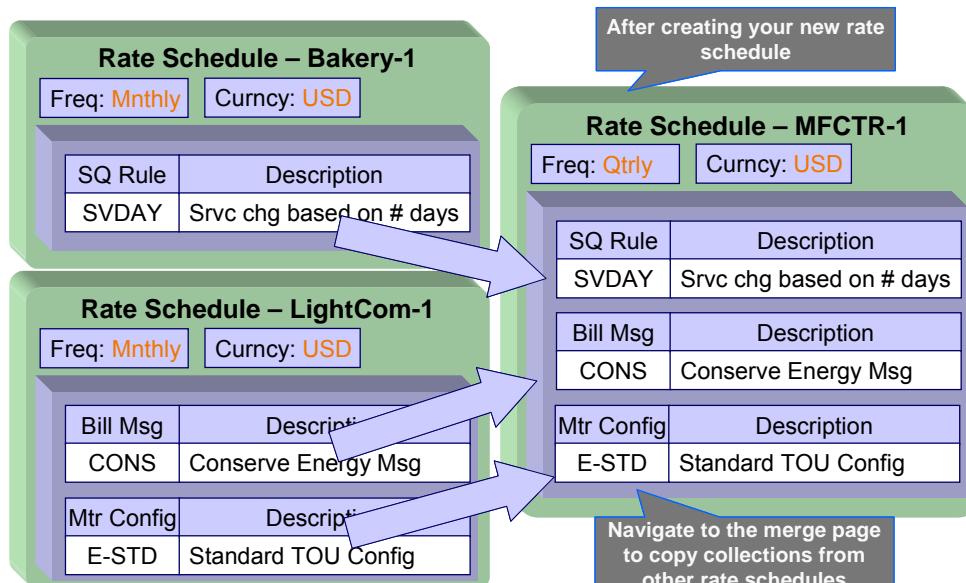
## Rate Schedule Merge

Use this page to modify an existing rate schedule by copying information from other rate schedules. This page may be used to copy records from the SQ rule, register rule, bill message and meter configuration type collections from one or more existing rate schedules to another. Some examples of when this page may be used are as follows:

- You wish to create a new rate schedule, which is similar to an existing rate schedule. Rather than copying all the information from the existing rate schedule and then removing the inapplicable components, this page may be used to selectively copy only the information applicable to the new rate schedule.
- Perhaps you offer a rate to your customer, which contains many optional components. Rather than building a generic rate schedule used by many customers, and using other logic to determine which options are applicable to which customer, you may choose to build a custom rate schedule for each customer, using only the components applicable to that customer. In this scenario, you may choose to create special 'mini' rate schedules, one for each of the various options. Then, you could use the rate schedule merge page to select the components applicable for the new custom rate schedule.

- Perhaps you are adding several new bill messages, which are applicable to multiple existing rate schedules. Once you have added the new messages to one rate schedule, you may find it easier to update the subsequent rate schedules by using the rate merge page to copy the messages across.

**Note.** The target rate schedule must exist prior to using this page. If you are creating a new rate, you must first go to the [Rate Schedule](#) page to add the new rate schedule and then navigate to the merge page to copy collection information.



**Duplicate versus Merge.** The [Rate Schedule](#) page has [Duplication](#) capability. You would duplicate a rate schedule if you want to a) create a new rate schedule AND b) populate it with all the information from an existing rate schedule. You would use the rate schedule merge page if you want to build a rate schedule using pieces of one or more rate schedules.

Open **Main Menu, Rates, Rate Schedule Merge** to open this page.

#### Description of Page

Select the **Original Rate Schedule** that is the target for merging the rate schedule collection information.

Select the **Merge From Rate Schedule** that is your template rate schedule to copy the collections from.

**Note.** You may only copy information from one Merge From rate schedule at a time. If you wish to copy information from more than one rate schedule, select the first Merge From rate schedule, copy the desired records, Save, then select the next Merge From rate schedule.

The left portion of the page will display any existing records in the collections for the original rate schedule. The right portion of the page will display the existing records in the collections for the Merge From rate schedule.

You may use the **Copy All** button to copy all the records in all the collections from the Merge From rate to the Original rate. If you do not choose to copy all, you may copy records individually as described below.

The left portion of the **SQ Rule** collection initially displays existing SQ rule records linked to the original rate schedule. In the **Merge Type**, you will see the word **Original**, for any of these records. The **Sequence** number and **SQ Rule** description are displayed. In the right portion of the collection, the existing records in the merge from rate are displayed initially.

The left portion of the **Register Rule** collection initially displays existing register rule records linked to the original rate schedule. In the **Merge Type**, you will see the word **Original**, for any of these records. The **Sequence** number and **Register Rule** description are displayed. In the right portion of the collection, the existing records in the merge from rate are displayed initially.

The left portion of the **Bill Messages** collection initially displays existing bill messages linked to the original rate schedule. In the **Merge Type**, you will see the word **Original**, for any of these records. The description of each **Bill Message** is displayed. In the right portion of the collection, the existing records in the merge from rate are displayed initially.

The left portion of the **Meter Configuration** collection initially displays existing meter configuration records linked to the original rate schedule. In the **Merge Type**, you will see the word **Original**, for any of these records. The description of each **Meter Configuration** is displayed. In the right portion of the collection, the existing records in the merge from rate are displayed initially.

The topics, which follow, describe how to perform common maintenance tasks:

## Contents

- [Removing A Row](#)
- [Adding A New Row](#)
- [Removing An Uncommitted Row](#)
- [Moving Rows Up and Down](#)

## Removing A Row

If you wish to remove a record linked to the Original rate schedule, press the “-“ button to the left of the record.

## Adding A New Row

You may move any of the records from the Merge From rate to the original rate 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

If you have copied a row across by mistake, you may remove it by clicking on the right arrow adjacent to the appropriate record.

## Moving Rows Up and Down

The **SQ Rule** and **Register Rule** grids contain sequence numbers that indicate the order in which the rules should be executed. You may modify the execution order using the up and down arrows.

You may also need to renumber the sequences so that your rules are executed in the proper order. The fields **Start from sequence number** and **Increment By** are used to define the sequence number to assign to the first rule and the subsequent values to assign to subsequent rules.

**Note** that once the record is saved, both register rules are marked as *Original*.

Refer to [Editable Grid](#) in the system wide standards documentation for more information about adding records to a collection by selecting from a list and repositioning rows within a grid.

## Setting Up Bill Factors

This section describes bill factors and how they are used to levy the many charges referenced in a rate.

**Warning!** Most bill factors exist to support calculations performed by rate components. This means that you must design your rate components before you can design your bill factors. We strongly recommend that you design “on paper” how every rate’s rate components looks before you attempt to set up bill factors.

### Contents

- [An Overview Of Bill Factors](#)
- [The Structure Of A Bill Factor](#)
- [An Illustration Of A Bill Factor And Its Characteristics](#)
- [Deriving / Passing In Characteristic Values](#)
- [Bill Factor Interval Values](#)
- [Bill Factor TOU Values](#)
- [Defining Bill Factors](#)
- [Defining Bill Factor Values](#)
- [Defining Interval Values](#)
- [Bill Factor Interval Values Query](#)
- [Upload Interval Values](#)
- [Defining Bill Factor TOU Values](#)

## An Overview Of Bill Factors

The primary reason why rate components exist is to define how much to charge for the services sold by your company. When you create rate components, you have four ways to indicate how much to charge:

- Specify the value directly in the rate component.
- Tell the rate component to use the value defined in a *bill factor*.

- Tell the rate component to use the value calculated by a “*for calculation purposes only*” rate component.
- Tell the rate component to call an algorithm to get the price.

The remainder of this section describes the bill factor approach. The other methods are discussed in [Designing Rate Components](#) and in [Rate Examples](#).

Bill factors are frequently used to specify a tax rate. For example, you could specify the current tax rate (say 6%) directly in every rate's rate components OR you could indicate every rate should levy the current state tax rate and have the system look up the applicable rate when it calculates bills. The latter approach involves using a bill factor to indicate that the amount to charge should be retrieved from someplace other than the rate component when a bill is calculated.

You can use bill factors to define any of the following types of charges:

- A flat amount. This would typically be used to define service charges, minimum charges, and asset charges.
- An amount for a given unit of measure. This would typically be used to define the price for measured consumption (e.g., the price per therm).
- A percentage. This would typically be used for tax rates and discount percents.

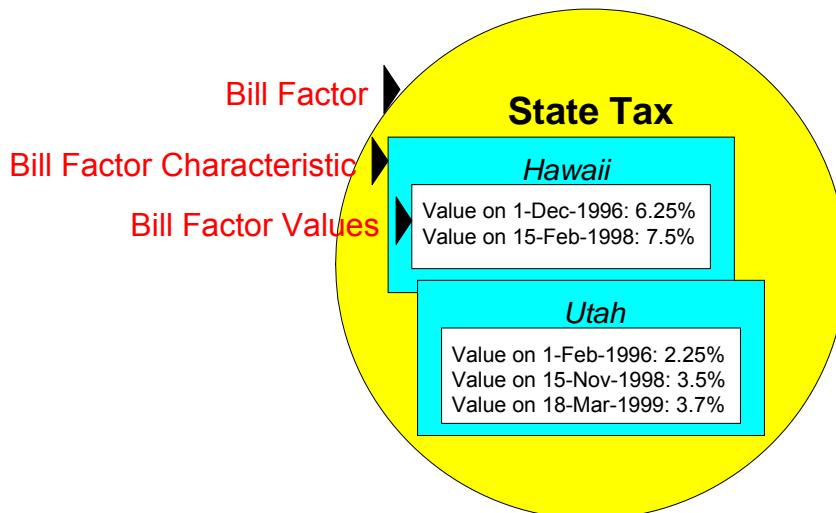
As a rule of thumb, you would use a bill factor (rather than specify the value in the rate component) when any of the following situations exist:

- The same charge exists in many rates. For example, if there are 3 rates that all contain the same service charge, it would make sense to use a bill factor to levy this charge rather than specify the same value on 3 rate components.
- The amount being charged is dependent on some physical characteristic of the customer's equipment. For example, if the price differs depending on whether the customer owns the transformer, you would use a bill factor to specify the price.
- The amount being charged is dictated by some external organization and therefore can change independently from the rate. For example, tax percentages and use fees are set by taxing authorities and can be changed at any time (and you don't want to change the rate when these charges change).
- The amount being charged varies depending on WHERE the customer lives. If you have a rate where the price per gallon differs depending on where the customer lives, you should use a bill factor to levy this type of charge.
- A charge (or discount) is only levied for a subset of customers for defined periods of time. For example, if you have a discount that is only given to customers who have had service for more than 2 years, you would use a bill factor to specify the discount percent.
- The price differs per customer. For example, you may have a rate for large gas customers where the price per therm is different for every customer. In this situation, you would use a bill factor to specify the price.
- The price differs based on the number of weeks of consumption. This typically occurs in a deregulated environment when the price charge is based on the weighted average spot market price in effect during the bill period. Refer to [Interval Prices Can Be Used For More Than Interval Prices](#) for more information.

**Warning!** As you can deduce, bill factors are extremely flexible. While the flexibility makes for a very powerful rating engine, it also presents you a dizzying array of options. We have discovered that to take advantage of the flexibility, you need to gain an overall understanding of the pricing and taxation provisions in ALL of your rates. Only then can you make logical and pragmatic decisions in respect of the number and type of bill factors.

## The Structure Of A Bill Factor

All bill factors share a common structure as illustrated below:



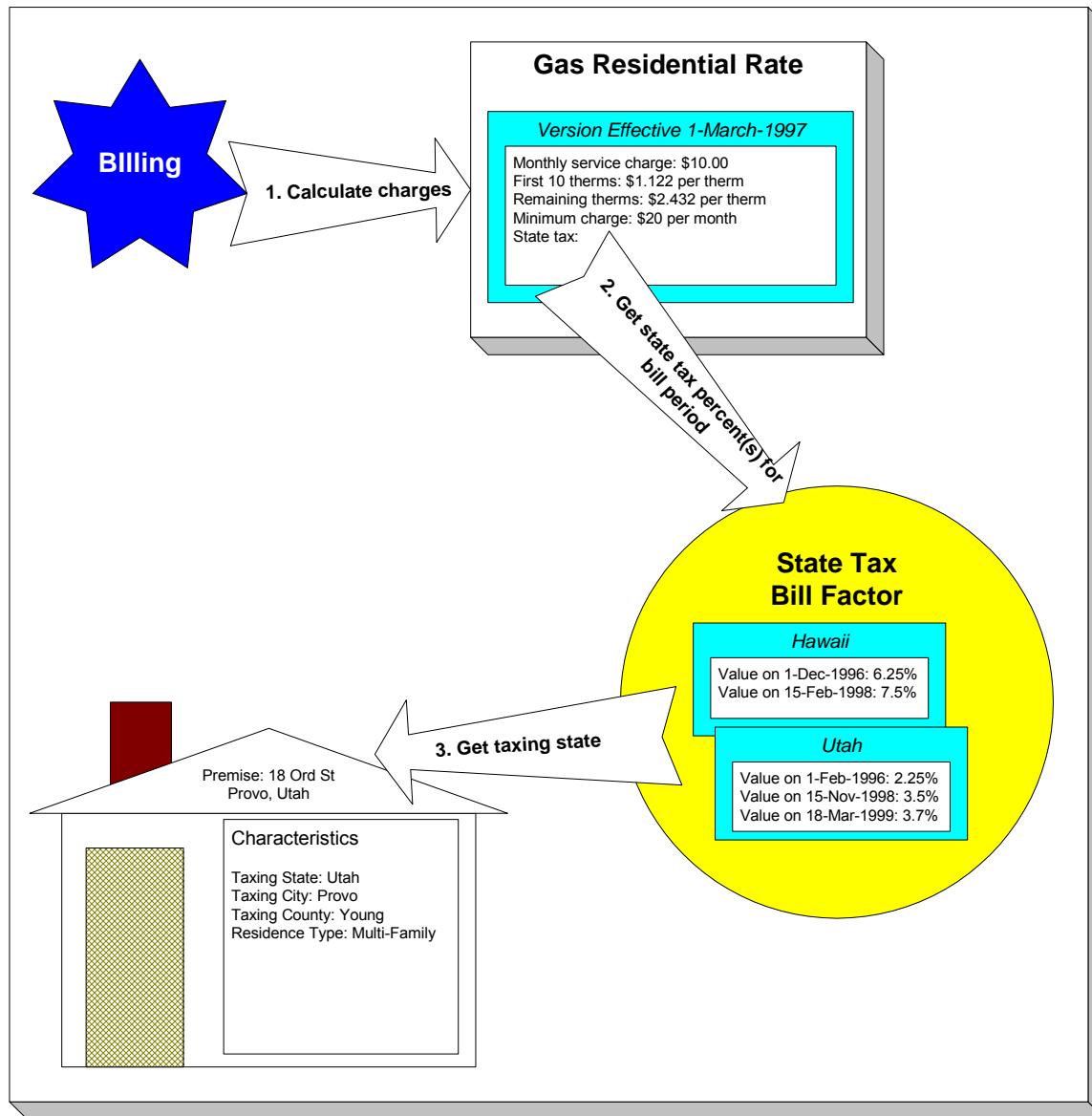
The above example shows a standard bill factor made up of the following tables:

- A **bill factor** contains descriptive information and attributes that control how the bill factor may be used in the system.
- All bill factors have the capability of having different values depending on some characteristic of the premise or service point. For example, a bill factor used to levy a state tax would have a different tax rate depending on the state in which the customer resides. A **bill factor characteristic** must exist for every state with a tax (states without a tax will not be linked to the state tax bill factor). At billing time, if the system cannot find a bill factor characteristic that corresponds with the customer's state of residence, the charge will be skipped.
- And finally, each bill factor characteristic may have many **bill factor values** over time. To continue with our state tax example, each state with a tax can have different tax rates over time. There are different ways in which a bill factor's values may be defined:
  - For bill factors whose values are applicable to many customers where only one value is effective on a given date, e.g., a city tax, the values are defined using bill factor values. Refer to [Defining Bill Factor Values](#) for more information.
  - For bill factors whose values are applicable to many customers where more than one value is effective on a given date, e.g., an interval price or a weighted average price, the values are defined on an interval value set. Refer to [Defining Interval Values](#) for more information about interval value sets.
  - For bill factors that define values for a collection of use codes for a given date, the values are defined using bill factor TOU values. Refer to [Bill Factor TOU Values](#) for more information about TOU values.

- For bill factors whose value differs per customer, the value is defined in the customer's service agreement. Refer to [Service Agreement – Contract Value](#) for more information about bill factors in contract terms.

## An Illustration Of A Bill Factor And Its Characteristics

The following picture illustrates how a bill factor and its characteristics are used to retrieve the relevant state tax at billing time:



The following points describe the above:

- At billing time, the billing engine sends a request to a rate to calculate the charges for a customer's consumption.

- The rate calculates charges without bill factors until it encounters the rate component used to levy state tax. This rate component references a bill factor. This means the rate must get the tax rate(s) in effect during the billing period from this bill factor.
- The state tax bill factor contains an attribute defining that the premise's taxing state characteristic controls the bill factor value. The bill factor therefore extracts the taxing state from the premise.

**Deriving characteristic values.** Rather than have the system extract the characteristic value from an entity, you can setup the system to derive the characteristic value when the rate is calculated. For example, if all of your customers are located in a single state, you may not want to maintain the taxing state on every premise. To do this, you could setup the rate to "hard code" a taxing state of say Hawaii. This is an advanced topic, but it may prove useful for your implementation. Refer to [Deriving / Passing In Characteristic Values](#) for the details.

- The premise returns the taxing state and the bill factor extracts the tax rate(s) in effect during the bill period and returns them to the rate.
- The rate applies the tax percents and returns the charges to billing.

**Some bill factors don't need a characteristic.** There are bill factors whose value does not differ based on a characteristic. For example, if your organization offers employee discounts, the discount percent is probably the same for all employees and therefore this bill factor doesn't need a characteristic to select the bill factor value at billing time. However, because of the relational design of the system, every bill factor value must reference both a bill factor and a characteristic value. Therefore, if you have bill factors whose value is not related to a characteristic, you must specify a characteristic type on the bill factor with a characteristic value that is exactly the same as the characteristic type. Refer to [Employee Discounts](#) for an example.

For more information about setting up characteristics, see [Setting Up Characteristic Types & Their Values](#).

## Deriving / Passing In Characteristic Values

**Warning!** This is an advanced, technical concept.

In [An Illustration Of A Bill Factor And Its Characteristics](#), we described how the system retrieves a characteristic value from a premise when a bill factor is encountered that uses a premise characteristic. This is a very common technique and is easy to understand. However, there are situations when an entity in Oracle Utilities Customer Care and Billing doesn't contain the appropriate characteristic value and therefore other techniques must be used. These techniques may not apply to your implementation, but they're worth understanding to help you form an intuitive understanding of rate application:

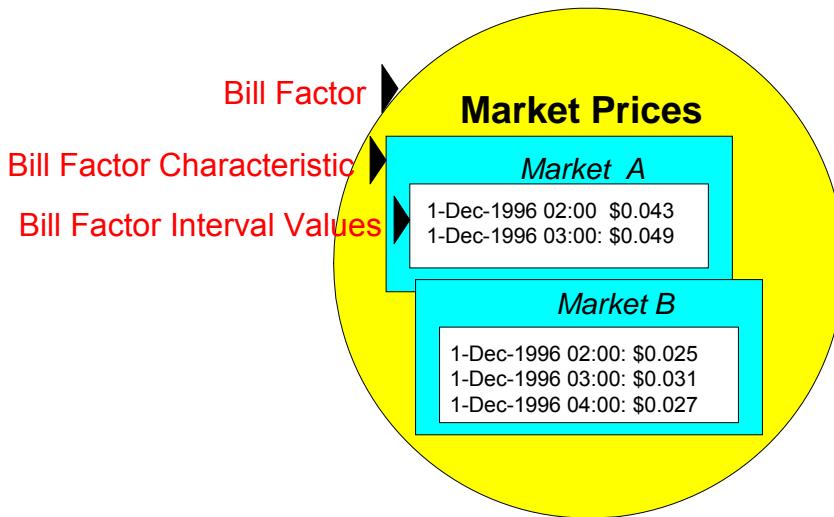
- There are implementations that only use the Oracle Utilities Customer Care and Billing rating engine. These implementations do not setup accounts, premises, etc. Rather, they simply setup their rate schedules, rate components, and bill factors and then call the rating engine from another application. This means the rate engine can't retrieve characteristic values from within Oracle Utilities Customer Care and Billing (because there are no objects on which the characteristic values can be defined). In this situation, the application that calls the rate engine must supply the characteristic types and values to be used when a bill factor requires this type of information.
- Assume you have a charge that is based on the account's customer class and the service agreement's revenue class. This type of charge's characteristic value is not extractable from a single entity. Rather, the system must construct the characteristic value real-time by concatenating information from the account and service agreement being billed. In this situation, you can use an SQ rule to derive this characteristic type and value. This SQ rule would create a characteristic type and value real time by concatenating the account's customer class and the service agreement's revenue class. This characteristic type and value only exists in memory while rate application executes. This is an example of the system deriving a characteristic value as opposed to it being passed in or retrieved from another object in the system.

When you setup a bill factor to use a characteristic whose value is derived or passed in, you need to define its characteristic source as **Characteristics Collection**. This source tells the system to extract the characteristic value from memory rather than extracting it from an object within the system.

**Note.** The various plug-in spots on a rate component have access to the **Characteristics Collection**. You might find this useful if you have a rate component that needs to calculate characteristic values for subsequent rate components.

## Bill Factor Interval Values

A bill factor may also contain interval values.



Interval values are typically used to record interval prices (i.e., prices that are applicable to short time intervals of say 15 minutes). Just as with simple bill factors, they may have characteristics defined. In the above example, the characteristic is the “Market” where the prices are defined. This bill factor probably indicates that the Service Agreement has a characteristic indicating the market from which it gets its prices. This enables you to use a common rate for customers who get their prices from different markets.

The application of these interval values requires the use of an algorithm. Depending on your business needs, there are many ways that this can be implemented.

- If your business uses the interval billing subsystem, then functionality is available for applying these interval values to interval quantities. Refer to [The Big Picture of Interval Billing](#) for more information.
- If your business does not use the interval billing subsystem, then there is still functionality available in rates for using interval values. See [Interval Values Can Be Used For More Than Just Interval Prices](#) for one example. Refer to [Pricing Interval Consumption Using Interval Prices](#) for another example.

## Contents

- [Interval Values Can Be Used For More Than Just Interval Prices](#)
- [Interval Data is Stored in Data Sets](#)

## Interval Values Can Be Used For More Than Just Interval Prices

The interval value transaction can be used for more than recording interval prices. One such purpose is to record rolling average prices. Let's use an example to make the point. Assume the following:

- You operate in a deregulated environment and part of the price charged to your customers is based on the spot-market price.
- Your customers don't have interval meters. Rather, they simply have meters that are read once a month. When you calculate a customer's bill, you must determine the single price to be applied to the total amount of consumption for the month.

- The price applied to a customer's bill is based on the number of weeks of consumption and the end date of the bill period. For example, a customer whose bill ends on 12-Sept-2001 with 3 weeks of consumption would probably have a different price than a customer whose bill ends on the same date but has 6 weeks of consumption. Why? Because the spot market price varies during the different billing periods.
- Different prices may be calculated based on the customer's load profile. This would mean that a residential customer whose bill ends on 12-Sept-2001 with 3 weeks of consumption would have a different price than a commercial customer whose 3 week bill ends on the same date. Why? Because residential customers tend to consume in different time periods than do commercial customers (and therefore different spot market prices were in effect).

If you assume the above, then you'd need the ability to record different prices based on:

- The bill end date
- The number of weeks / days in the billing period
- The customer's load profile

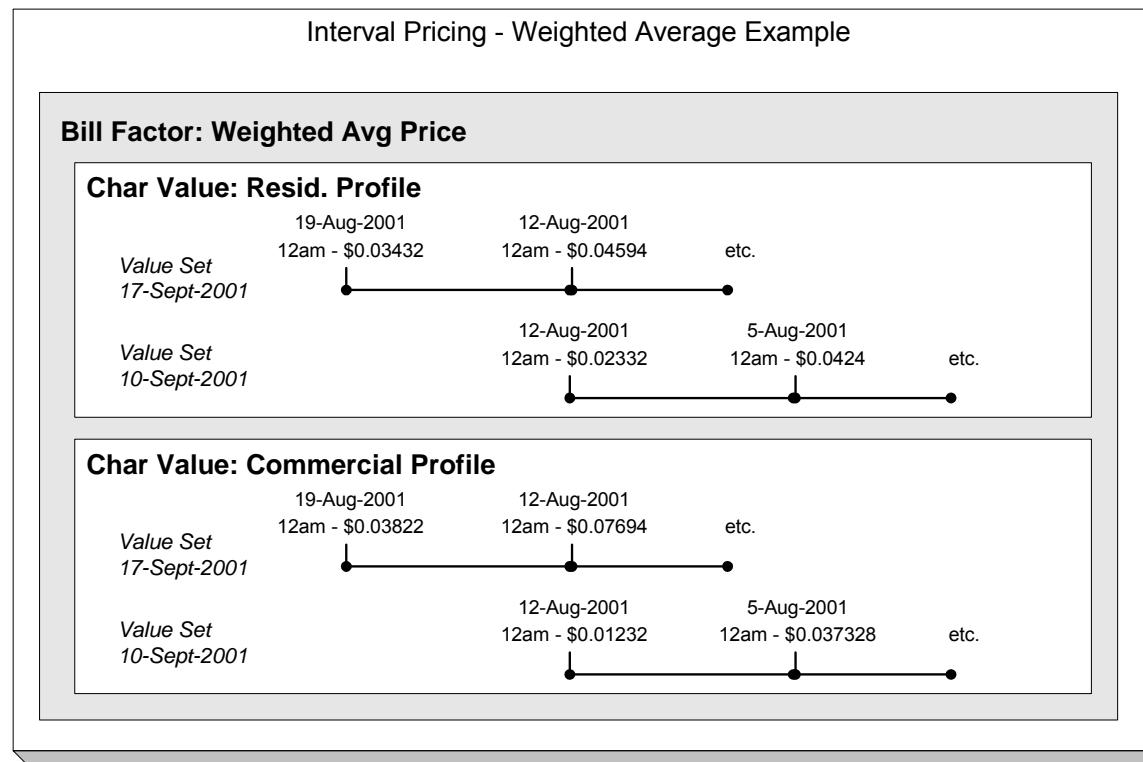
You can use the interval pricing objects to record this information. You do this as follows:

- Create a characteristic type called LOADPROF. Associate with it characteristic values of RESID and COMMERCIAL.
- Create a bill factor called WAP and have it reference the LOADPROF characteristic type.
- Enter an interval pricing value set for each bill end week cutoff date. And for each interval pricing value set, add an interval value for each bill start week cutoff date.

The following table will help visualize an interval pricing value set for the bills ending the week of 17-Sept-2001:

<b>Bill Factor: WAP – Weighted Average Price</b> <b>Char Value: RESID – Residential Load Profile</b> <b>Interval Value Date / Time: Bills ending the week of 17-Sept-2001</b>		
<b>Bill Period</b>	<b>Bill Start Date</b>	<b>Weight Avg Price</b>
4 Week Weighted Average PX Price	19-Aug-2001	0.03432
5 Week Weighted Average PX Price	12-Aug-2001	0.04594
6 Week Weighted Average PX Price	5-Aug-2001	0.02302
7 Week Weighted Average PX Price	29-July-2001	0.06039
8 Week Weighted Average PX Price	22-July-2001	0.02981
9 Week Weighted Average PX Price	15-July-2001	0.01921
10 Week Weighted Average PX Price	8-July-2001	0.05010
11 Week Weighted Average PX Price	1-July-2001	0.060929
12 Week Weighted Average PX Price	24-June-2001	0.03939

Below, we've illustrated how interval pricing information might look for a bill factor that holds the weighted average prices.



Refer to [Pricing Non-Interval Consumption Using A Weighted Average Price](#) for how a rate component value algorithm makes use of the above information when a bill is calculated for a customer.

## Interval Data is Stored in Data Sets

Interval data in the system will have many different origins.

- It may be contract-related data, which is setup at the beginning of the contract or contract year
- It may be data, which is interfaced from an external source periodically

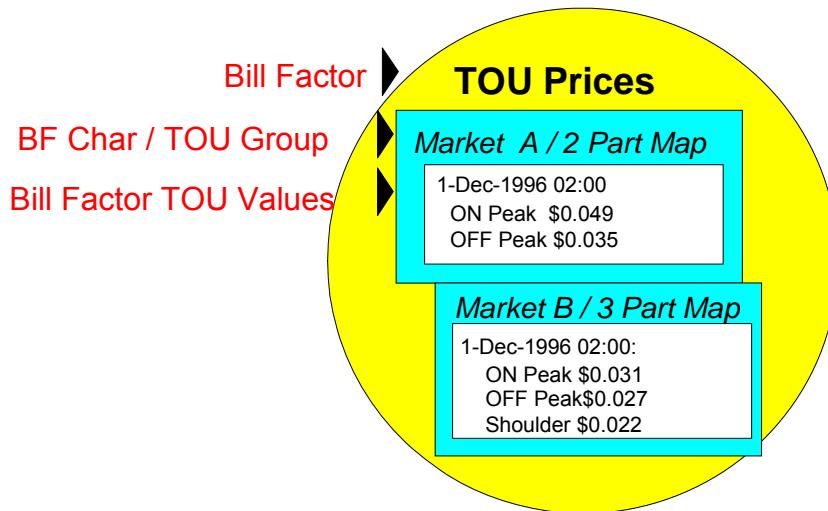
Regardless of the origin of the data, all data will have similar characteristics

- Data will be received by the system, or created by the system periodically. The system will store whatever new data needs to be stored.
- The period represented by the batch of new data is variable.
- New data may be an update to previously received data

Refer to [Interval Quantities are Stored in Data Sets](#) for information about how interval data is stored. This same model is used for bill factor interval data. There is no difference.

## Bill Factor TOU Values

A bill factor may also contain values, which differ by a TOU code. This enables you to set up prices that differ by TOU or to set up contract values, which differ by TOU.



Bill Factor TOU values are typically used to record prices, which vary by time of use code. TOU prices are used in conjunction with time of use mapping. The customer would typically have interval quantities, which are mapped, using a TOU Map, into a collection of time of use quantities. Defining a bill factor with prices per TOU allows you to set up a single rate component to apply charges to these quantities, resulting in multiple bill calculation lines.

Just as with simple bill factors, TOU bill factors may have characteristics defined. In the above example, the characteristic is the “Market” where the prices are defined. This bill factor probably indicates that the Service Agreement has a characteristic indicating the market from which it gets its prices. This enables you to use a common rate for customers who get their prices from different markets.

In addition to defining the characteristics, TOU bill factors must also indicate the TOU Group. The TOU group is a collection of time period definitions, which make up a TOU map. This enables you to define different prices for On Peak, when it is part of a 2 part TOU map versus when it is part of a 3 part TOU map.

Refer to [Grouping of TOU Codes](#) for more information.

The TOU prices are defined for the combination of characteristic and TOU Group. In the examples below, a bill factor's characteristic is State. Also note that the effective date of the prices are the same for all time of use codes. If the price of any TOU value changes, then a new set of prices for all TOU codes must be entered.

<b>TOUs in Group 2</b>	<b>California</b>
<b>Effective 1 Jul 1999</b>	
On Peak/Summer	\$0.98
Off Peak/Summer	\$0.74
On Peak/Winter	\$0.84
Off Peak/Winter	\$0.63

<b>TOUs in Group 3</b>	<b>California</b>
<b>Effective 1 Jul 2000</b>	
On Peak/Spring	\$1.03
Off Peak/Spring	\$0.74
On Peak/Autumn	\$0.92
Off Peak/Autumn	\$0.63

<b>TOUs in Group 2</b>	<b>Oregon</b>
<b>Effective 1 Jul 1999</b>	
On Peak/Summer	\$0.98
Off Peak/Summer	\$0.74
On Peak/Winter	\$0.84
Off Peak/Winter	\$0.63

The application of these time of use values requires the use of an algorithm. If your business uses the interval billing subsystem and uses time of use mapping and pricing, then functionality is available for applying these values to interval quantities, which have been mapped into time of use quantities. Refer to [The Big Picture of Interval Billing](#) for more information.

## Contents

- [TOU Values Can Be Used For More Than Just Prices](#)
- [TOU Values May Differ By Customer](#)

## TOU Values Can Be Used For More Than Just Prices

TOU values can be used for more than recording prices. Many contract-based values may also be recorded per TOU. For example, a contract for an interval billing customer may designate a hedge cover or a minimum demand value or a contract demand value. Although you may have interval quantity curves to define these values, that would be difficult to define and maintain. A more common scenario would be to define a TOU Map and to define these values for each TOU. A TOU Bill Factor could be used to define these values.

## TOU Values May Differ By Customer

In the same way that simple bill factors may be used to define customer specific prices and values, TOU bill factors may also be used to define customer specific prices and values. Refer to [Customer Specific TOU Values](#) for more information.

## Defining Bill Factors

You create a bill factor for every type of charge whose value is not specified directly in a rate component.

You use two windows to set up a bill factor. On the first, you define general information describing how the bill factor is used in the system. On the second, you define every characteristic value for which the bill factor has a value.

## Contents

- [Defining General Bill Factor Information](#)
- [Setting Up Bill Factor Characteristics](#)

### Defining General Bill Factor Information

You begin to define a bill factor by selecting **Main Menu, Rates, Bill Factor**.

#### Description of Page

Enter a unique **Bill Factor** code and a **Description** for the bill factor.

Use **Bill Factor Type** to indicate if this bill factor uses **Regular**, **Interval** (i.e., real time) or **Time of Use** values. If you choose **Regular**, this bill factor's values are specified on [Bill Factor Value – Main](#) or on [Service Agreement – Contract Values](#). If you choose **Interval**, this bill factor's values are specified on [Defining Interval Values](#). If you choose **Time of Use**, this bill factor's values are specified on [Bill Factor TOU Values](#) or on [SA Interval Billing Info - TOU Contract Values](#).

Enter a **Currency Code** to define the currency in which the bill factor's prices are denominated. This field will be gray if the bill factor is referenced on a rate component.

**Note.** Bill factors referenced on a rate component must be denominated in the same currency as is the rate.

Select the **Value Type** to define the kind of value maintained by this bill factor. The Value Type options are:

- |                   |   |
|-------------------|---|
| <i>Charge</i>     | The value maintained in this bill factor is a fixed amount (e.g., a flat charge). |
| <i>Percentage</i> | The value maintained in this bill factor is a percentage.                         |
| <i>Unit Rate</i>  | The value maintained in the bill factor is a price per service quantity.          |

This field will be gray if the bill factor is already referenced on a rate component or if the bill factor already has values.

Turn on **Error If No Value** if the system should create a billing error when a bill factor value doesn't exist for the characteristic value linked to the customer. Confused? To clarify this, you need to be aware that there are different types of bill factors:

- Those that should have a value for every customer whose rate references the bill factor. A bill factor used to levy the relevant state tax is a good example (assuming every state has a tax). For this type of bill factor, if the system cannot find a bill factor value for the taxing state, it must generate a billing error because data is missing from the system. You would turn on **Error if No Value** for this type of bill factor.
- Those that only have a value when the customer has a characteristic value that levies a charge. A bill factor used to levy the relevant city tax is a good example (because not all cities levy taxes). For this type of bill factor, if the system cannot find a bill factor value for the city in which the customer lives, the rate component will be skipped and no billing error will be generated. You would turn off **Error if No Value** for this type of bill factor.

Bill factors used to levy taxes may or may not be eligible for tax exemptions; it's up to the respective taxing authority. Customers who are eligible for the tax exemption will have their tax obligation reduced (or nullified) at billing time. For bill factors that are eligible for tax exemptions, turn on **Tax Exemption**. If a service agreement is fully or partially exempt for the taxes defined by a given bill factor, the bill factor and the exemption information for the customer must be indicated on the Service Agreement page. This field will be gray if the bill factor is referenced as being tax exempt in a service agreement.

**Tax Exemptions limited to Apply To Rate Components.** Although bill factors marked as eligible for tax exemptions may be referenced on several different types of rate components, only [Apply To](#) rate components include the logic to check for tax exemption eligibility.

For more information about how to make a customer tax exempt, refer to [Service Agreement – Tax Exemptions](#).

Some types of bill factors do not have the same value for all customers. Rather, their value may be unique for every customer or it may be unique for some customers and common for others. For example, you may have rates where the price per kilowatt hour is defined for every customer using the rate. You may have other rates where certain customers have their own price per kilowatt hour defined and other customers who use the value defined on the bill factor. For any type of bill factor where a customer may have their own unique price, turn on **Value In Contract Term**. For any service agreement that has a unique value for a given bill factor, the bill factor and the service agreement's value must be indicated on the [Service Agreement](#) page. When applying the rate for a bill factor of this type, the system will first check for the existence of a contract value for the service agreement and if one is not found, it will use the value on the bill factor. This field will be gray if a contract term referencing this bill factor exists. Time of Use bill factors may follow this same logic. Customers may override their time of use values by using a contract term.

**Note.** The **Error if No Value** switch also affects what happens if there is no contract term for the bill factor. If the error switch is on, a billing error will be generated. If the error switch is off, no billing error will be generated and the rate component will be skipped.

For an example of a bill factor whose value is specified in a contract term, refer to [Customer Specific Prices](#). For more information about how to specify a bill factor value in a customer's contract, refer to [Service Agreement – Rate Info](#) and [Service Agreement - TOU Contract Values](#).

Contract riders are used when the charge associated with the bill factor is only applicable to some customers for some period of time. For example, if a bill factor is used to provide a discount during the first 3 months of service, then a contract rider would be used. Turn on **Contract Rider Applicability** if the bill factor's applicability is contingent on the existence of a contract rider. This field will be gray if a contract rider referencing this bill factor exists on a service agreement.

For an example of a bill factor whose value is dependent on a contract rider, refer to [Employee Discounts](#). For more information about how to establish a contract rider for a customer, refer to [Service Agreement – Contract Riders](#).

The proration information tells the system how to handle changes to bill factor values and their optional contract riders that may occur during a customer's bill period. Turn on **Allow Proration** if changes to the bill factor's value and/or contract riders should be prorated. If the **Bill Factor Type** is not specified as **Interval** and if changes should **not** be prorated, indicate the **Rate Selection Date Algorithm** to use when the system retrieves bill factor values or determines if contract riders are applicable. Click [here](#) to see the algorithm types available for this system event.

If you have sub service agreements, turn on **Use Sub SA Contract** terms if billing should access contract riders, values, tax exemptions from the sub SA rather than the master. If this switch is off, the system will use the master SA's riders, contract value, and tax exemptions when it creates bills for a sub service agreement. Refer to [Sub Service Agreements](#) for more information.

If the **Contract Management** module is not [turned off](#), which includes [umbrella agreement management](#), and you have checked either Tax Exemption, Value in Contract Term or Contract Rider Availability, indicate the **Terms of Service Usage**. Valid values are **Check SA Only**, **Check TOS Only** and **Check TOS First, then SA**. Refer to [Overriding Rate Terms](#) for details about how to set this value based on your business needs.

### Bill Factor Tree

The right half of this page is dedicated to a tree that shows the bill factors characteristics and their respective values (for **Regular** bill factors) or interval value Sets (for **Interval** bill factors). You can use this tree to both view high level information about these objects and to transfer to the respective page in which an object is maintained by using the available Go To buttons.

### Where Used

Many types of Rate Components may use a Bill Factor to define the price being charged. Refer to [Rate Component – Main Information](#) for more information.

The **BILPC-TOTAL** algorithm type that is used to levy late payment charges references the bill factor that defines the late payment charge percent. Refer to [SA Type – Main](#) for more information about this algorithm.

A Service Agreement references Bill Factors as follows:

- When a service agreement's **Apply To** rate component references tax exemptable bill factors AND the customer is tax exempt, the service agreement will define the bill factors from which the customer is exempt. Refer to [Service Agreement – Tax Exemptions](#) for more information.
- When a service agreement's rate references bill factors whose prices are customer-specific (i.e., specified on a contract term), the service agreement will contain prices for these bill factors. Refer to [Service Agreement – Contract Values](#) for more information.
- When a service agreement's rate references bill factors that are contingent on contract riders, the service agreement will define the rider period (if any) for these bill factors. Refer to [Service Agreement – Contract Riders](#) for more information.

Register rules using the **Bill factor based UOM conversion** algorithm reference a bill factor to define the conversion amount. Refer to [CCF To Therm Conversion](#) and [Gas Pressure Factor Adjustments](#) for examples.

## Setting Up Bill Factor Characteristics

After defining general information about a bill factor, open **Main Menu, Rates, Bill Factor** and navigate to the **Bill Factor Characteristic** tab to define characteristic values for which a charge is applicable. For example, a bill factor used to levy the applicable city tax would require the definition of every city that levies a tax.

**Note.** If all values for a bill factor are specified in contract terms, you do not need to set up Bill Factor Characteristics.

### Description of Page

Enter a **Characteristic Type** to indicate the type of characteristic that controls how the bill factor's value is defined. This field will be gray if Bill Factor Characteristics are specified in the following grid.

The **Characteristic Source** defines where the system retrieves the characteristic value at billing time. The Characteristic Source options are:

<i>Account</i>	The characteristic type's value will be retrieved from the service agreement's account.
<i>Characteristic Collection</i>	Refer to <a href="#">Deriving / Passing In Characteristic Values</a> for an explanation of this value.
<i>Main Person</i>	The characteristic type's value will be retrieved from the main person linked to the service agreement's account.
<i>N/A</i>	The characteristic type's value is the same for all customers using the rate. If this option is used, you must still choose a Characteristic Type. This Characteristic Type should have a single Characteristic Value. The ID of the characteristic value must be identical to the ID of the characteristic type. Refer to <a href="#">Employee Discounts</a> for an example.
<i>Premise</i>	The characteristic type's value will be retrieved from the service agreement's characteristic premise.
<i>SA</i>	The characteristic type's value will be retrieved from the service agreement.
<i>SP</i>	The characteristic type's value will be retrieved from the customer's service point. If multiple service points are linked to the service agreement, one will be chosen at random.

For more information about characteristics, see [Setting Up Characteristic Types & Their Values](#) and [An Illustration Of A Bill Factor And Its Characteristics](#).

Use the add and remove buttons to define the **Characteristic Values** that are relevant for the bill factor's Characteristic Type. For example, if the bill factor is used to levy a city tax, you would define the city's that have a city tax in the above collection.

Use the **Minutes Per Interval** if your Bill Factor is of type **Interval**. This defines, for each characteristic value, the number of minutes expected between each record of interval data.

Use the **External ID** if the values for this characteristic are interfaced to the system from an external source and you want to record an identifier for the source.

If your Bill Factor is of type **Interval** and the [installation record](#) indicates that [seasonal time shift](#) is required, then you must enter the appropriate **Seasonal Time Shift** record applicable for the interval data. Please take special note of the issue described in the [Evenly Sized Intervals](#) section.

For more information, refer to [Seasonal Time Shifts](#).

**Note.** If the bill factor's value is the same for all customers, you still must link a "dummy" characteristic value to the bill factor. In other words, you'll have to create a characteristic type called "n/a" and define for it a characteristic value of "n/a". Then, on the bill factor, you reference the "n/a" characteristic type and on the above window, you indicate a characteristic value of "n/a".

## Defining Bill Factor Values

After the bill factor and its characteristic values are defined, you define the price / percent and GL distribution associated with each bill factor characteristic. The topics in this section describe the pages used to set up this information.

### Contents

- [Bill Factor Value - Main](#)
- [Bill Factor Value - GL Distribution](#)

### Bill Factor Value - Main

After the bill factor and its characteristic values are defined, select **Main Menu, Rates, Bill Factor Values** to define the price / percent applicable to each bill factor characteristic.

**Important!** You do not set up Bill Factor Values for bill factors whose values are specified in an interval value data set or if all the values are customer specific and are defined as contract terms. Refer to [Defining Interval Values](#) for more information about interval value sets. Refer to [Service Agreement – Contract Value](#) for more information about bill factors in contract terms.

#### Description of Page

Enter the **Effective Date** and respective **Value** for the bill factor and characteristic value. The first date should be the earliest date on which the bill factor's charges are effective. Subsequent effective dates and values are required whenever the value changes.

**Default note.** A **Value** of 0 will be defaulted.

### Bill Factor Value - GL Distribution

Select **Main Menu, Rates, Bill Factor Value** and navigate to the **GL Distribution** tab to define the GL account affected by the bill factor characteristic.

**Important!** If you do not specify this information for a bill factor characteristic, the distribution code defined on the rate component that references the bill factor will be used.

#### Description of Page

If the bill factor characteristic's charges are booked to a different GL account depending on the characteristic value, specify the appropriate **Distribution Code** for each **Revenue Class**.

**Optional information.** Specifying distribution codes on a bill factor is optional. You would only do this if the GL distribution code differs based on the characteristic value (e.g., the city tax payable GL account differs depending on the city). If you do not enter the above information, the distribution code specified in the rate component that references the bill factor will be used.

## Defining Interval Values

Open **Main Menu, Rates, BF Interval Values** to maintain interval values.

**Interval values must reference a bill factor.** You must set up a bill factor before you can define interval prices. Refer to [Setting Up Bill Factors](#) for more information.

### Description of Page

**Interval Value Set** is a concatenation of important information about the data set. The **Interval Value Set ID** is a system-assigned random number that stays with an interval value set for the life of the system. This value is assigned when the set is first added.

All interval pricing information must reference a predefined **Bill Factor, Characteristic Type** and **Characteristic Value**. Use the **Bill Factor Characteristic** search button to select all three values. Refer to [Setting Up Bill Factors](#) for more information.

Use **Set Date / Time** to define the effective date of the set of prices or values defined in the following grid. This is considered a server date and will follow the seasonal time shifting as defined on the base time zone. Refer to [Seasonal Time Shifts](#) for more information.

Use the **Start Date / Time** to indicate the starting point for the display of the interval data.

At least one interval data record must exist in the collection of interval data. For each interval data record, the **Interval Value Date / Time** and the **Interval Value** are captured. This date is considered a logical date and will follow the seasonal time shifting as defined on the bill factor / characteristic. Refer to [Seasonal Time Shifts](#) for more information.

If seasonal time shift records are linked to both the base time zone and the bill factor/characteristic, the **Seasonal Time Shift** associated with your bill factor/characteristic is displayed. Refer to [Setting Up Bill Factor Characteristics](#) for more information.

If seasonal time shift records are linked to both the base time zone and the bill factor/characteristic, the **Seasonal Time Shift Remark** indicates whether the data is displayed and should be entered in legal time or standard time. Refer to [Interval Time Display](#) for more information. If the data is displayed in legal time use the **Show in Standard** button to toggle to standard time. If the data is displayed in standard time use the **Show in Legal** button to toggle to legal time. If either the bill factor/characteristic or the base time zone do not indicate a seasonal time shift record, then the toggle button will not be visible and data is displayed and expected in standard time.

### Where Used

Various rate component algorithms make use of interval pricing information.

## Bill Factor Interval Values Query

This query shows the most recent interval values for an interval value bill factor/characteristic and a time period as of a given time.

Open **Main Menu, Rates, BF Interval Values Query** to display interval bill factor values.

### Description of Page

Information describing the **Bill Factor**, **Characteristic Type** and **Characteristic Value** whose data you wish to query is displayed.

Use the **Start Date / Time** and **End Date / Time** if you want to limit the query to a specific time period.

Use the **Set Date / Time** to indicate the “as of what date” to use for retrieving interval data. Only data sets with a set date/time prior to this date/time will be accessed.

The **Selection Type** is used to determine what data to show

#### History

This option will display all bill factor interval data for the Bill Factor/Characteristic and time period, if specified. It will include data in data sets of any status value.

#### Most Recent Complete Data

This will find the most recent bill factor interval value data for each interval from completed data sets only.

Once you have selected your criteria, use the search button to display the data. The interval value data collection will contain the following information:

The **Interval Value Date/Time** and **Interval Value** for each interval is displayed. Adjacent, information about the data set to which this interval is linked is displayed, including the **Set Date / Time** and **Interval Value Set ID**.

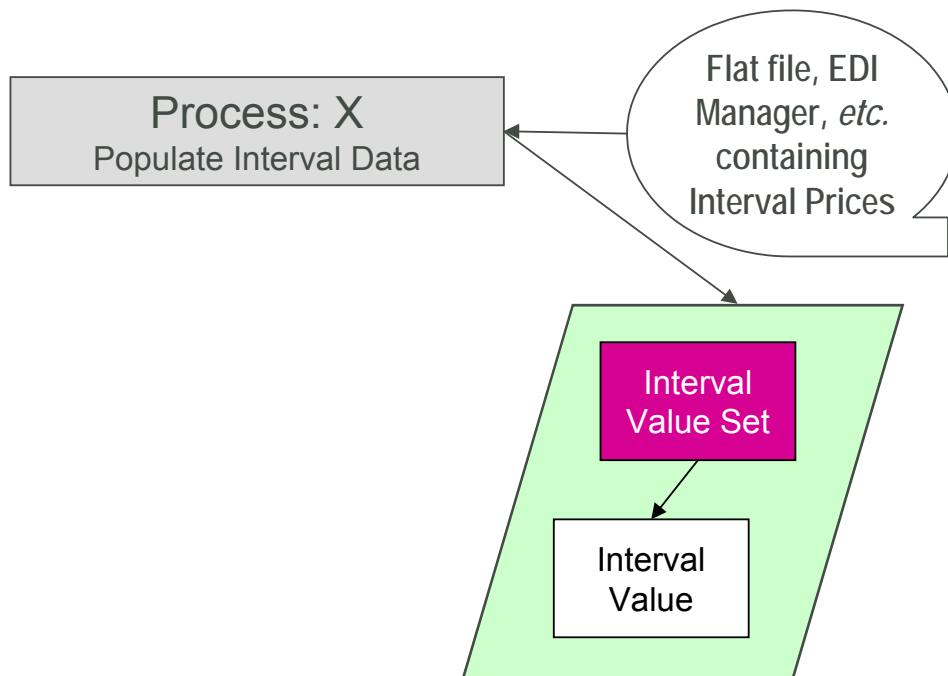
If seasonal time shift records are linked to both the base time zone and the bill factor/characteristic, the **Seasonal Time Shift** associated with your bill factor/characteristic is displayed.

The **Seasonal Time Shift Remark** indicates whether the data is displayed and should be entered in legal time or standard time. Refer to [Interval Time Display](#) for more information. If the data is displayed in legal time use the **Show in Standard** button to toggle to standard time. If the data is displayed in standard time use the **Show in Legal** button to toggle to legal time.

**Note.** If seasonal time shift information is missing for the bill factor/characteristic or for the base time zone, then data is always displayed and expected in standard time.

## Upload Interval Values

The following diagram illustrates the processes involved in the uploading of interval values into the system.



The topics in this section describe how these processes work.

### Contents

- [Process X - Populate Interval Data](#)
- [Interval Value Set](#)
- [Interval Value](#)
- [Interval Value Set Key](#)

### Process X - Populate Interval Data

Process X refers to the mechanism used by your organization to populate the Interval Value tables. Process X will have the following responsibilities:

- It will determine the appropriate Bill Factor and Characteristic to which this data belongs. Recall that the bill factor / characteristic may reference an external id.
- It will create a new Interval Value Set for this bill factor characteristic with the current date and time
- If the interval data is from a different time zone, or is not in standard time for the base time zone, the time of each interval must be converted to the base time. Refer to [Time Zone and Time Changes](#) for more information.
- All prices being uploaded for the same bill factor and characteristic should be inserted into the Interval Value table and should reference the same Interval Value Set
- Inserts a record in the interval value set key table.

The topics in this section describe the tables to populate.

## Interval Value Set

You must create a data set for each batch of records being uploaded in the system for the same bill factor / characteristic. The name of this table is [CI\\_INTV\\_VAL\\_SET](#). The following table describes each column on this table.

Column Name	Length	Req'd	Data Type	Comments
INTV_VAL_SET_ID	12	Y	A/N	<p>This is a random unique key. This key is referenced on the interval value record.</p> <p>It's suggested that this is a random number. If your process that inserts records on this table is capable of calling a COBOL routine, call CIPZKEYG and it will supply a 12 digit random number for you.</p>
BF_CD	8	Y	A/N	<p>This along with CHAR_TYPE_CD and CHAR_VAL indicate the bill factor / characteristic to which this interval value set belongs.</p> <p>Refer to <a href="#">Setting Up Bill Factors</a> for more information.</p>
CHAR_TYPE_CD	8	Y	A/N	<p>This along with BF_CD and CHAR_VAL indicate the bill factor / characteristic to which this interval value set belongs.</p> <p>Refer to <a href="#">Setting Up Bill Factors</a> for more information.</p>
CHAR_VAL	16	Y	A/N	<p>This along with BF_CD and CHAR_TYPE_CD indicate the bill factor / characteristic to which this interval value set belongs.</p> <p>Refer to <a href="#">Setting Up Bill Factors</a> for more information.</p>
SET_DTTM	26	Y	DateTime	<p>This is the date/time that the data set was received. This must be populated with the current date / time. This is considered a server date and will follow the seasonal time shifting as defined on the base time zone.</p> <p>Refer to <a href="#">Seasonal Time Shifts</a> for more information.</p> <p>Refer to <a href="#">Algorithms Find the Most Up To Date</a> for information about why this date/time is important.</p>

## Interval Value

Each piece of interval data for the same bill factor/characteristic will be stored as an Interval Value record linked to the newly created data set. The name of this table is [CI\\_INTV\\_VAL](#). The following table describes each column on this table.

Column Name	Length	Req'd	Data Type	Comments
INTV_VAL_SET_ID	12	Y	A/N	This is a foreign key reference to the Interval Value Set that groups the interval values together. This should be your newly created id from above.
INTV_VAL_DTTM	26	Y	DateTime	This is the date and time of the interval value. This must be stored relative to the base time zone. Refer to <a href="#">Time Zone and Time Changes</a> for more information.
INTV_VAL	18.7	Y	N	This is the value (price) for the interval.

### Interval Value Set Key

When a system-generated key value is assigned to a record, you need to also store the key value in a [key table](#) that corresponds to the record's database table. The name of this table is [CI\\_INTV\\_VAL\\_SET\\_K](#). The following table describes each column on this table.

Column Name	Length	Req'd	Data Type	Comments
INTV_VAL_SET_ID	12	Y	A/N	This should be your newly created <a href="#">Interval Value Set</a> ID from above.
ENV_ID	6	Y	N	This is the environment id defined on the <a href="#">installation options</a> .

## Defining Bill Factor TOU Values

Open **Main Menu, Rates, BF TOU Values** to maintain time of use bill factor values.

**TOU Bill Factor values must reference a bill factor and characteristic.** You must set up a bill factor and its characteristics before you can define TOU values. Refer to [Setting Up Bill Factors](#) for more information.

### Description of Page

The **BF Characteristic TOU Group** search button gives you the ability to search for existing TOU BF values for a Bill Factor, Characteristic and TOU Group combination.

The **BF Characteristic** search button gives you the ability to search for a Bill Factor, Characteristic combination to which the TOU values will be linked.

Once a bill factor and characteristic combination are chosen, the system displays the values of the **Bill Factor, Characteristic Type, Characteristic Value and Value Type**.

Choose the **TOU Group** to determine the collection of time of use codes to which prices will be defined.

Indicate the **Effective Date** of the collection of **Time of Use** codes and their **Values**.

### Where Used

Various rate component algorithms make use of TOU value information. This information is also available for any type of algorithm that may need to use the information, for example interval data derivation algorithms.

## Defining Rate Versions

After defining general information about a rate on [Rate Schedule Maintenance](#), you must link to it a rate version. The rate version defines the effective date of the calculation rules defined in its rate components. A rate schedule will have multiple rate versions if its calculation rules (i.e., its rate components) change over time.

**A switch on the rate schedules controls what happens if multiple rate versions are effective during a bill period.** When the system creates a bill segment for a service agreement, it checks if multiple rate versions are in effect during the bill period. If so, it uses the [rate schedule's Allow RVs Proration](#) switch and [RV Selection Date](#) flag to determine if it should prorate the various rate versions or if it should pick one of the rate versions effective during the bill period.

After a rate version exists, you add rate components to it using [Rate Component Maintenance](#) and/or [Rate Version Merge](#). After you have added all necessary rate components, don't forget to return to [Rate Version - Main](#) and change the state of the rate version to [\*\*Finished\*\*](#) (otherwise, it cannot be used by billing).

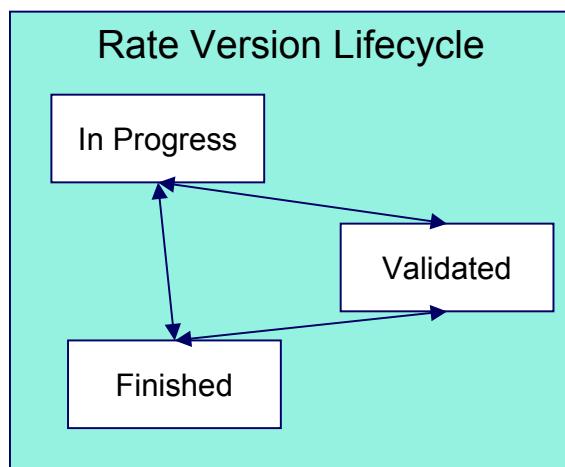
The topics in this section describe how to setup a rate version.

### Contents

- [Lifecycle of a Rate Version](#)
- [Rate Version - Main](#)
- [Rate Version - Bill Print Info](#)

## Lifecycle of a Rate Version

The following diagram shows the possible lifecycle of a rate version (you'll notice that you can change a rate version from any state to any state).



A rate version is in the **In Progress** state while you build its rate components. While a rate version is in this state, the system does not perform cross validation between rate components as they are added / changed / deleted using [Rate Component Maintenance](#) and/or [Rate Version Merge](#). However, the system will perform inter-field validation applicable to each type of rate component. Rate versions in this state may not be used by [Rate Check](#), nor will they be used up by the billing process.

You transition a rate version to the **Validated** state when you're ready to check it using [Rate Check](#), but you don't want billing to use it to calculate charges. When a rate version is transitioned to this state, the system performs cross validation between its rate components. If any validation errors are found, an error message is displayed and the status remains **In Progress**. If all validation checks are passed, the rate version status will become **Validated**. We'd like to stress that rate versions in this state may be used by [Rate Check](#), but will not be used by billing.

You transition a rate version to the **Finished** state when it can be used by billing to calculate charges. All of the cross validation between rate components will be performed when transitioning to this state.

**Finished and Validated states also affect rate component maintenance.** If a rate version is **Finished** or **Validated**, the system performs internal consistency checks every time a rate component is added, changed or deleted using [Rate Component Maintenance](#) and/or [Rate Version Merge](#). If you don't want these checks to be performed while you're working on a rate version's rate components, change its state to **In Progress**.

## Rate Version - Main

Use **Main Menu, Rates, Rate Version** to maintain a rate version.

### Description of Page

**Rate Version** contains basic information about the rate version. This information only appears after the rate version has been added to the database.

**Rate Schedule** defines the rate to which the rate version is linked. **Effective Date** defines the date on which the rate version's rate components become effective. These fields are the unique identifier of the rate version. Both fields become protected after the rate version is added to the database.

**Multiple rate versions may be prorated.** When the system creates a bill segment for a service agreement, it checks if multiple rate versions are in effect during the bill period. If so, it uses the [rate schedule's Allow RVs Proration](#) switch and **RV Selection Date** flag to determine if it should prorate the various rate versions or if it should pick one of the rate versions effective during the bill period.

The **Rate Version Status** indicates the current state of the rate version. Refer to [Lifecycle of a Rate Version](#) for more information.

**Warning!** Billing ignores rate versions that are not in the **Finished** status. This means that if you create a new rate version and forget to finish it, billing will use the prior rate version when it calculates the customers' bills.

Specify the description of the rate version in **Description on Bill**. This description is saved on bill calculation details created using this rate version. The following table illustrates the substitution variable(s) that may be used to substitute variables into the description during billing:

Substitution Character	What It Does
%D	<p>Replaces the substitution character with message 10101 from message category 4. This message is maintained in the standard message table.</p> <p>We recommend this message be set to "(prorated for %1 of %2 days)" where %1 will be substituted with the number of days of prorated consumption charged on the bill line and %2 will be the "normal" days associated with the rate's prices. This message will only be substituted when the rate version is prorated.</p>

The following are examples of the content of common **Description on Bills** along with how the line would look on the bill:

Description on Bill	How It Looks On A Bill
Gas service %D	Gas service (prorated for 10 of 30 days)

The [tree](#) that follows contains a summary of the rate version's rate components. You can use this tree to both view high-level information about these objects and to transfer to the respective page on which an object is maintained.

**Note.** The duplicate action in the action button bar enables you to duplicate a rate version and its rate components. Refer to [Duplicate Button](#) in the system wide standards document for more information.

## Rate Version - Bill Print Info

This page allows you to maintain the bill print information for all of the rate version's rate components. Note, you can also maintain this information for individual rate components on [Rate Component – Main](#).

### Description of Page

The grid contains an entry for each rate component linked to the rate version. The following information is displayed in the grid:

**Sequence** uniquely identifies the rate component. It also controls the order in which the rate component is processed when the rate version is processed by billing.

**Description** is the rate component's description.

**Description on Bill** is the verbiage that appears on the bill line generated for this rate component. You can dynamically substitute variables into this description (e.g., price and quantity) when billing calculates the rate component's charges. Refer to [How To Use Description on Bill](#) for a discussion of the various substitution variables.

Turn on the **Print** switch if you want the bill line to appear on the customer's bill.

**Some bill lines don't print.** You may wonder why we give you the ability to not print a bill line. The reason is because ALL bill lines are shown on Bill Segment – Calculation Details. However, only those lines marked as Print are presented to the customer. This way, if you want to suppress some calculation details, you can, but still show the details to the customer service rep.

The **Print If Zero** switch is only enabled if the **Print** switch is turned on. It allows you to indicate if a bill line should print when the bill line's value is zero:

- Turn this switch on if the bill line should print when the calculated value is zero.
- Turn this switch off if the bill line should not print when the calculated value is zero.

**Note.** The rate component results in the creation of a bill segment calculation line, but an indicator exists to tell the extract program that this should not be included. This is true, except in the following cases when the system does not create a zero bill segment calculation line:

1. **Apply To** rate components where the cross referenced amount is zero
2. **Calculation Algorithm** rate components that result in a zero amount
3. **Summary, Minimum, Maximum** and **Exact Charge** rate components that do not apply
4. **Stepped** rate components where both the low and high boundaries equal zero

### How To Use Description on Bill

Enter the verbiage to appear on the bill segment in the **Description On Bill**. Turn on **Print** if the verbiage should appear on the printed bill.

**Suppressing calculation details on the printed bill.** Only rate components used to calculate intermediate values are typically suppressed on a bill. For these types of rate components, we recommend using **Description on Bill** to show users how the line was calculated, but turn off the **Print** switch. This way, users will see the calculation details on the [Bill Segment – Calc Lines](#) window, but the customer won't see it on the bill.

Many lines that appear on a bill do not contain static verbiage. Rather, the printed lines contain information from the calculation process. For example, the bill line for state tax would typically contain the tax percent.

You control exactly what is substituted in the bill line by entering one or more of the following substitution variables in the **Description on Bill** field:

Substitution Character	What It Does
%A	<p>Replaces the substitution character with the amount of the bill line. Please be aware of the following:</p> <ul style="list-style-type: none"> <li>• If the rate component isn't designated as being "for calculation purposes only" (FCPO), all trailing zeroes up to the decimal position of the rate's currency will be removed. If the rate component is designated as being "for calculation purposes only" (FCPO), the number of decimal places as defined in the rate component's precision will be shown.</li> <li>• All leading zeroes will be removed unless the value is between 1 and -1 in which case a leading 0 will appear before the decimal place.</li> <li>• The Symbol on the Currency control table will prefix the value.</li> <li>• The position of the negative sign and currency symbol is dependent on the display profile of the user who generate the bill (or who submitted the batch billing process).</li> </ul>
%B	<p>Replaces the substitution character with the begin/start date covered by the bill calculation line. If the charge is proratable, %B may be used in conjunction with %E to display the period covered by the bill calculation line.</p>
%C	<p>Replaces the substitution character with the number of items being billed. Leading zeroes are suppressed.</p>
%D	<p>Replaces the substitution character with message 10101 from message category 4. This message is maintained in the standard message table.</p> <p>We recommend this message be set to "(prorated for %1 of %2 days)" where %1 will be substituted with the number of days of prorated consumption charged on the bill line and %2 will be the "normal" days associated with the rate's prices. This message will only be substituted when the rate component is prorated. Note, neither %1 nor %2 are required.</p> <p>Either this character OR %F should be used when you desire to substitute the price / percent into the bill line.</p> <p>If the charge is proratable, %D should be used in conjunction with %R to print an appropriate message when the charges are prorated.</p>
%E	<p>Replaces the substitution character with the end date covered by the bill calculation line. If the charge is proratable, %E may be used in conjunction with %B to display the period covered by the bill calculation line.</p>
%F	<p>Replaces the substitution character with message 10102 from message category 4. This message is maintained in the standard message table.</p>

	<p>We recommend this message be set to "(prorated by %1)" where %1 will be substituted with the factor used to prorate the charge. This message will only be substituted when the rate component is prorated. Note, %1 is not required.</p> <p>If the charge is proratable, %F should be used in conjunction with %I to print an appropriate message when the charges are prorated.</p>
%I	<p>Replaces the substitution character with the value specified in the rate component / bill factor.</p> <p>If the value is a percent,</p> <ul style="list-style-type: none"> <li>• all leading and trailing zeroes will be suppressed,</li> <li>• and a "%" will be suffixed to the resultant value.</li> </ul> <p>If the value is not a percent,</p> <ul style="list-style-type: none"> <li>• all trailing zeroes up to the decimal position of the rate's currency will be removed,</li> <li>• and all leading zeroes will be removed unless the value is between 1 and -1 in which case a leading 0 will appear before the decimal place,</li> <li>• and note that the position of the negative sign and currency symbol is dependent on the display profile of the user who generate the bill (or who submitted the batch billing process).</li> </ul> <p>Either this character OR %R should be used when you desire to substitute the price / percent into the bill line. Using %I will cause the value as specified in the rate component to be substituted regardless of whether the charges are prorated.</p> <p>If the charge is proratable, %I should be used in conjunction with %F to print an appropriate message when the charges are prorated.</p>
%O	Replaces the substitution character with the description of the time-of-use (TOU) code being billed on this line
%Q	<p>Replaces the substitution character with the service quantity being billed. Leading zeroes are suppressed. The decimal places shown are specified in a field on the Unit Of Measure control table.</p> <p>If you wish to suppress trailing zeroes after the decimal, you must set up the <i>Allow Consumption Zero Suppression In Bill Description</i> option type on the <i>Financial Transaction Options Feature Configuration</i>.</p>
%R	<p>Replaces the substitution character with the value specified in the rate component / bill factor.</p> <p>Either this character OR %I should be used when you desire to substitute the price / percent into the bill line. Using %R causes the prorated price / percent to be printed in the bill line (%I causes the price / percent as specified in the rate to be printed during proration situations).</p> <p>If the charge is proratable, %R should be used in conjunction with %D to print an appropriate message when the charges are prorated.</p>
%S	<p>Replaces the substitution character with the summed amount. All trailing zeroes up to the decimal position of the rate's currency will be removed. All leading zeroes will be removed unless the value is between 1 and -1 in which case a leading 0 will appear before the decimal place. The Symbol on the Currency control table will prefix the value. If the value is negative, a "-" will prefix the value.</p> <p>This character should only be used with Minimum, Apply To, and Summary rate components.</p>
%T	<p>Replaces the substitution character with message 10103 from message category 4. This message is maintained in the standard message table.</p> <p>We recommend this message be set to "(%1% exempt)" where %1 will be substituted with the tax</p>

	exemption percent as specified in the service agreement. This message will only be substituted when the rate component's bill factor is eligible for tax exemptions AND the customer is exempt from the tax. Note, %1 is not required.
Lines that don't print	If the bill line is not printed (as defined by the Print switch), message 10104 from message category 4 will be appended to the bill line. This message is maintained in the standard message table.  We recommend this message be "NOT SEEN BY CUSTOMER!" This message appears in the bill line that appears in the billing windows but does not appear on the customer's bill.

The following are examples of the content of common **Description on Bills** along with how the line would look on the customer's bill:

Description on Bill On Rate Component	How It Looks On A Bill
%C lamp pole(s) at %I per pole %F	1,234 lamp pole(s) at \$5.00 per pole (prorated by .3333)
%C lamp pole(s) at %R per pole %D	1,234 lamp pole(s) at \$1.50 per pole (prorated for 10 of 30 days)
First %Q kWh at %I per kWh %F	First 100.32 kWh at \$0.12349 per kWh (prorated by .3333)
First %Q kWh at %R per kWh %D	First 100.32 kWh at \$0.12349 per kWh (prorated for 10 of 30 days)
%Q kWh at %I per kWh	100.32 kWh at \$0.12349 per kWh
State sales tax %R %T	State sales tax 6.25% (50% exempt)
Minimum charge of %I %F	Minimum charge of \$10.00 (prorated by .3333)
Subtotal of %S	Subtotal of \$1,123.23
Charges for %O: %Q kW at %I per kW	Charges for Off Peak Winter: 234.43 kW at \$0.05 per kW

The prorated references will only appear if the rate component's charges are prorated due to a short or long bill period.

## Designing Rate Components

Rate components are the fundamental building blocks of your bill calculation algorithms. All of the other rate and bill factor set up tasks are done in preparation of the creation of these records. This means that you must design your rate components before you can design bill factors, register rules, SQ rules, SQ identifiers, eligibility rules. We strongly recommend that you design "on paper" how every rate component looks on every rate schedule before you attempt to set up any of the rate control tables.

All of the calculation details and financial information that appear on a bill segment are derived using information on rate components. If you want to see the results of what rate components do, refer to [Bill Segment – Calc Lines](#).

**Warning!** There are innumerable ways to design rate components for a given rate. 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 rate components, we encourage you to gain intuitive understanding of these options by using the system to prototype the alternatives.

The topics in this section provide background information that will facilitate the construction of your rate components.

## Contents

- [Rate Component Design Methodology](#)
- [Rate Component Rounding](#)
- [The Big Picture Of Rate Component Eligibility Rules](#)

## Rate Component Design Methodology

Designing rate components is an iterative process. Over time, you will develop intuitive skills that will allow you to avoid some iterations. However, when you're starting out, we recommend you follow these steps to design your rate components:

- 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. If the rate has optional charges (e.g., employee discounts or charges that only apply after the customer has consumed a large volume), make sure your examples encompass every possible scenario.
- Next, look at the bill lines and ask yourself what are the variables that cause each line's charges to be calculated the way they are. Consider the following examples:
  - "Monthly service charge: \$50.00". This is a simple service charge that appears regardless of how much the customer consumed. This line doesn't really have any variables that affect its amount. Rather, the amount is a constant value of \$50.00.
  - "1,250 kWh at \$0.07892". This charge is based on how much the customer consumed. The variables involved are the number of kilowatt-hours (kWh) and the price per kWh.
  - "1,000 20K Lumen Mercury Vapor Lamps at \$20.00 per lamp". The variables involved are the number of devices (also called "items") used by the customer and the price per item.
  - "Minimum monthly charge of \$100.00". This is a charge that only appears if the total of the prior lines is less than \$100.00. The variables involved in this line are a little complicated because they must be calculated at billing time by adding up several other lines and comparing them to the minimum charge amount. This means the variables are the total of prior lines and the minimum charge amount.
  - "1.25% County Surcharge". This charge is similar to the minimum charge in that it is calculated by adding up one or more prior lines and applying a percentage to the sum. This means the variables are the total of prior lines plus the tax percentage.
- After you've determined all of the potential bill lines you can start designing your rate components. Typically you create one rate component for every line that can appear on the customer's bill. When you create a rate component, you will categorize it as one of the following types:

**Flat Charge** This type of rate component is used to create bill lines that levy service charges and fees that aren't based on how much the customer uses. The monthly service charge shown above would be levied using this type of rate component.

**Service Quantity** This type of rate component is used to create bill lines that levy charges based on some type of consumption. The kWh charge shown above would be levied using this type of rate component.

Frequently the amount consumed is measured by a meter's registers and sometimes it's calculated using Service Quantify (SQ) Rules linked to the rate and sometimes it's derived based on the type of item (e.g., for lamps). Regardless of how the amount consumed is

calculated, the charge for the consumption amount is levied using this type of rate component.

You'll find it helpful to differentiate your service quantity rate components into two categories – stepped and non-stepped. Stepped charges are those where there's a different price for different ranges of consumption (e.g., the first 50 therms are \$0.43 a therm, anything in excess of 50 therms is charged at \$0.71 per therm). We recommend doing this because a) you frequently show a subtotal of the steps on the customer's bill, b) steps, by definition, mean you have several rate components that are dependent on each other, and c) they frequently involve complex calculations in order to define the step boundaries.

<b>Item Type</b>	This type of rate component is used to create bill lines that levy charges based on the number and type of items used by a customer. The system keeps track of items on the service point(s) linked to the customer's service agreement. The lamp charge shown above would be levied using this type of rate component.
<b>Apply To</b>	This type of rate component is used to create bill lines that levy charges based on the amount calculated on other bill lines. The county surcharge shown above would be levied using this type of rate component.
<b>Summary</b>	This type of rate component is used to create a "subtotal" on the bill. It exists purely for aesthetic purposes.
<b>Minimum Charge</b>	This type of rate component is used to create bill lines that levy charges only when the sum of previously calculated lines is less than the minimum charge amount. The minimum monthly charge shown above would be levied using this type of rate component.  Note that if the values being compared are negative values, the comparison is NOT done on the absolute values, but rather on the actual values. For example, imagine you have a minimum discount of \$ -2.00 and previous rate components have calculated a discount of \$ -1.00. You want a rate component to create a bill line for \$ -1.00 to apply a further discount. A minimum charge rate component will not work in this case because -1 is considered more than -2. For this business scenario you should use a maximum charge rate component.
<b>Maximum Charge</b>	This type of rate component is used to create bill lines that levy charges only when the sum of previously calculated lines is more than the maximum charge amount.  Note that if the values being compared are negative values, the comparison is NOT done on the absolute values, but rather on the actual values. For example, imagine you have a maximum discount of \$ -2.00 and previous rate components have calculated a discount of \$ -3.00. You want a rate component to create a bill line for \$1.00 to reduce the discount. A maximum charge rate component will not work in this case because -3 is considered less than -2. For this business scenario you should use a minimum charge rate component.  Note – these types of rate components are rare. You tend to only see them in deregulated markets where customers have agreed to pay the market rate but want a cap on the maximum they'll pay each bill period.
<b>Exact Charge</b>	This type of rate component is used to force a bill to add up to a given amount (regardless of how much the bill would be based on earlier rate components).  Note – these types of rate components are rare. You tend to only see them in deregulated markets during the transition period between regulated rates and complete deregulation. In this situation, you show on the bill what the charges would be, given a deregulated environment, but then adjust the bill back to an amount equal to the regulated rate. This charge is called a Competition Transition Charge (CTC) in California.
<b>Calculation Algorithm</b>	This type of rate component enables you to produce bill calculation lines based on logic in an algorithm that you supply. Use this rate component when none of the other rate component functions will provide you with the logic you require.

<b>Interval Pricing</b>	This type of rate component is used to apply interval prices to interval quantities. See <a href="#">Interval Pricing Rate Application</a> for more information.
<b>TOU Pricing</b>	This type of rate component is used to map and/or price interval quantities into time of use quantities given a TOU map. See <a href="#">Time of Use Mapping and Pricing</a> for more information.

- For each rate component, determine if the customer must meet some form of eligibility criteria before the rate component is applied. For example, you may have a rate component that calculates an employee discount. This rate component may have eligibility criteria that requires the person to have a given characteristic value before the rate component is applied. Refer to [The Big Picture of Rate Component Eligibility](#) for more information.
- For each type of rate component (except summary), determine how you'll define its unit rate / percent / flat amount. You have four choices:
  - Specify the value directly in the rate component.
  - Tell the rate component to use the value defined in a *bill factor*.
  - Tell the rate component to use the value calculated by a “*for calculation purposes only*” rate component.
  - Tell the rate component to call an algorithm. The algorithm will return the value.

The following tables provide guidance in respect of which of the methods to use:

<b>Specify value in the rate component</b>	You'd specify the value in the rate component when the value in the rate is specific to that rate and is applied to all customers regardless. For example, if you see a provision in the rate that says to charge \$0.05674 per kWh, you'd specify 0.05674 directly in the rate.
<b>Use a bill factor to define the value</b>	<p>The following paragraphs describe when you'd use a bill factor to define a unit rate / percent / flat amount.</p> <p>The same charge exists in many rates. For example, if there are 3 rates that all contain the same service charge, it would make sense to use a bill factor to levy this charge rather than specify the same value on 3 rate components.</p> <p>The amount being charged is dependent on some physical characteristic of the customer's equipment. For example, if the price differs depending on whether the customer owns the transformer, you would use a bill factor to specify the price.</p> <p>The amount being charged is dictated by some external organization and therefore can change independently from the rate. For example, tax percentages and use fees are set by taxing authorities and can be changed at any time (and you don't want to change the rate when these charges change).</p> <p>The amount being charged varies depending on WHERE the customer lives. If you have a rate where the price per gallon differs depending on where the customer lives, you should use a bill factor to levy this type of charge.</p> <p>The price differs per customer. For example, you may have a rate for large gas customers where the price per therm is different for every customer. In this situation, you would use a bill factor to specify the price.</p>
<b>Use a “for calculation purposes only” rate</b>	This is complicated to explain and is not used very often, but it is very powerful and handles some very complicated algorithms. It works like this – you calculate the price / percent / flat charge on another rate component (marked as “ <i>for calculation purposes only</i> ” so it won't affect the bill amount), and then reference it on the “real” rate component. See the <a href="#">Calculated</a>

*component*    [Minimum Charges](#) example for an illustration of when this method is useful.

*Call an algorithm*    This is complicated to explain and is not used very often, but it is very powerful and handles some very complicated examples. It works like this – you call an algorithm and it calculates the price / percent / charge.

One example that would require the use of an algorithm would be when the price differs based on the total amount consumed. For example, if the customer uses less than 10 CCF, the price is \$1.50; whereas if the customer uses more than 10 CCF, the price is \$2.00.

We supply an example of one such algorithm in the base package. If your rates require additional algorithms, you will have to develop additional algorithms. Refer to [How to set up service quantity rate components](#) under [Rate Component – Main Information](#) for more information about this type of algorithm.

Please note that this scenario could also be implemented without invoking an algorithm by setting up two rate components that have eligibility rules to indicate their required respective consumption levels. Refer to [The Big Picture of Rate Component Eligibility](#) for more information.

- After you've categorized the bill lines into the various types and know how the price is defined, you determine how each line's charges are reflected in your general ledger. You will need to work with your accounting department as they will tell you the exact revenue, expense, and liability accounts that will be affected by the charges.
- Finally, read the provisions of your rate (the legal wording) and make sure you haven't left out something. This may involve having to discuss confusing provisions with your legal department.

At this point, you're ready to start entering your rate components.

**Examples.** Refer to [Rate Examples](#) for many sample rate components.

## Rate Component Rounding

The topics in this section describe how to control how each rate component's final value is rounded.

### Contents

- [Rounding Precision Is Defined On A Rate Component](#)
- [Rounding Method Is Defined On A Rate Component](#)
- [Rounding And FCPO Rate Components](#)
- [Interim Rounding For Apply To Rate Components](#)
- [Rounding At The End](#)

### Rounding Precision Is Defined On A Rate Component

A rate schedule references a currency code. A rate's currency code defines its maximum number of decimal places. These decimal places, in turn, control the smallest unit to which most rate components can be rounded (i.e., the rate component's precision). For example,

- If a rate's currency has 2 decimal places, the smallest rounding precision that most of its rate components can have is 0.01. In other words, the smallest unit to which most dollar-based rate components can be rounded is to the cent. Note, whether you round up / down / to the nearest is discussed below.
- If a rate's currency has no decimal places, the smallest rounding precision that most of its rate components can have is 1.00.

The system supports the notion of a rate component having a greater rounding precision than its currency. For example, a rate's currency may support 2 decimal places, but you can setup rate components to have a rounding precision of 0.05.

The reason that we underlined the word most in the previous paragraphs is that these rules do not apply to "for calculation purposes only" (FCPO) rate components. FCPO rate components don't contribute real amounts to a bill (they exist to calculate intermediate results that are used by later rate components) and therefore they can support a rounding precision of up to 0.00001 (5 decimal places).

## Rounding Method Is Defined On A Rate Component

When you setup a rate component you must define the rounding method. You are given the following choices:

- ***Always round up.*** This rounding method rounds a rate component up to the nearest value (as controlled by the rate component's precision). For example, if a rate component's precision is 0.01 and a value of 0.011 is calculated, the rate component's final value would be rounded up to 0.02.
- ***Always round down.*** This rounding method rounds a rate component down to the nearest value (as controlled by the rate component's precision). For example, if a rate component's precision is 0.01 and a value of 0.019 is calculated, the rate component's final value would be rounded down to 0.01.
- ***Round to the nearest value.*** This rounding method rounds a rate component to the nearest value (as controlled by the rate component's precision). For example, if a rate component's precision is 0.01 and a value of 0.019 is calculated, the rate component's final value would be rounded up to 0.02. Whereas, if a value of 0.012 is calculated, the rate component's final value would be rounded down to 0.01.

## Rounding And FCPO Rate Components

As described above, for calculation purpose only (FCPO) rate components don't contribute real amounts to a bill. FCPO rate components exist to calculate intermediate results that are used by later rate components. We therefore allow FCPO rate components to have a rounding precision greater than the rate's currency code to a maximum of 0.00001 (5 decimal places).

Please be aware of the following:

- As described under [How To Use Description on Bill](#), you can indicate that the final amount of a bill line should be substituted into the bill line's description by using the %A substitution variable in the rate component's **Description On Bill**. If you do this for an FCPO rate component, the system will show the number of decimal places as dictated by the FCPO's precision.

- While rate application calculates bill lines, the system maintains each rate component's value in memory. For FCPO rate components, this means that precisions greater than .01 are available during rate application. For example, you could indicate an FCPO rate component has a precision of .0001 and the resultant value will be maintained in memory and is available to other rate components while rate application executes.

When rate application completes, the FCPO's value will be either saved in the bill line's amount or discarded as per the value of the rate component's **FCPO retention rule**. If you indicate that the FCPO amount should be retained on the bill line and its precision is greater than 2 decimal places, the FCPO amount will be rounded to two decimal places before it's saved on the database. Keep in mind that this could result in an inconsistency if you used the %A substitution variable in the rate component's **Description On Bill** as the value substituted into the bill line will have a greater precision.

### Interim Rounding For Apply To Rate Components

The calculations for Apply To rate components involve several steps with interim values stored along the way. For every interim value that is stored, rounding rules are applied. The following steps describe the interim rounding rules:

- If the value type is not **Unit Rate**, the amounts for all the "cross-reference" rate components are added together and stored in the database as the Base Amount. The base amount is stored with a precision of .01 and it is rounded to the nearest value based on the currency's precision.
- The rate component calculates its charge and keeps the result in a temporary field with a precision of .00001.
- The remaining steps depend on whether or not a tax exemption is applicable
  - If NO tax exemptions are applicable, the temporary field is rounded based on the rate component's rounding rules to produce the final amount.
  - If there are tax exemptions, the temporary field is rounded to the nearest .01 decimal precision prior to applying the tax exemptions. When the tax exemption is applied, the result is rounded according to the currency precision and stored as the Exempt Amount. The system calculates the final amount by subtracting the exempt amount from the rounded temporary amount and then applying the rate component's rounding rules.

### Rounding At The End

The following is an example of a simple bill segment with three bill lines.

Bill Segment - SA1		
Line Description	Amount	GL Impact
Usage charge	\$501.00	Revenue \$501.00
Tax	\$5.01	Tax payable \$5.01
Total	\$506.05	Rounding \$0.04

The first two bill lines are calculated using simple rate components that have a precision of .01. The last bill line is special, it rounds the bill up to the next highest 0.05 (this is an example from a country that doesn't have cents, their smallest coin is 0.05). The following would be necessary to calculate the last bill line:

- The rate component would be an exact charge rate component
- Its value (i.e., the exact charge) would be calculated using a value calculation algorithm that sums the first two rate components and rounds them up to the nearest .05. Refer to [RCVL-RNDXRF](#) for a value calculation algorithm that can perform this function.

An alternative approach is to do the following:

- Use a summary rate component to calculate the exact charge. This rate component would require the following characteristics:
  - FCPO
  - Non printing
  - Rounding type would be round up
  - Precision would be 0.05
- Reference the summary rate component as the exact charge rate component's value

## The Big Picture Of Rate Component Eligibility Rules

You can use eligibility rules to define the criteria when a rate component should be applied (or skipped). For example,

- You may have a rate component that calculates an employee discount. This rate component may have eligibility criteria that require the person to have a given characteristic type and value (this assumes you use a person characteristic to identify employees) before the rate component is applied.
- You may have a rate component that calculates a surcharge if the customer uses 25% more than they used in the same period in prior year.
- ...

**Eligibility rules are optional.** If you put eligibility rules on a [rate component](#), the rate component will be skipped or applied as per the instructions in the rules. If you don't put eligibility rules on a rate component, the rate component will be executed (i.e., rate components are eligible by default).

The topics in this section describe eligibility rules.

### Contents

- [A Rate Component Is Eligible By Default](#)
- [Criteria Groups versus Eligibility Criteria](#)
- [Defining Logical Criteria](#)
- [Examples Of Rate Component Eligibility Rules](#)

### A Rate Component Is Eligible By Default

If you don't want to setup eligibility rules, you don't have to. If you don't specify eligibility rules on a rate component, it's eligible (i.e., it's going to be processed by rate application).

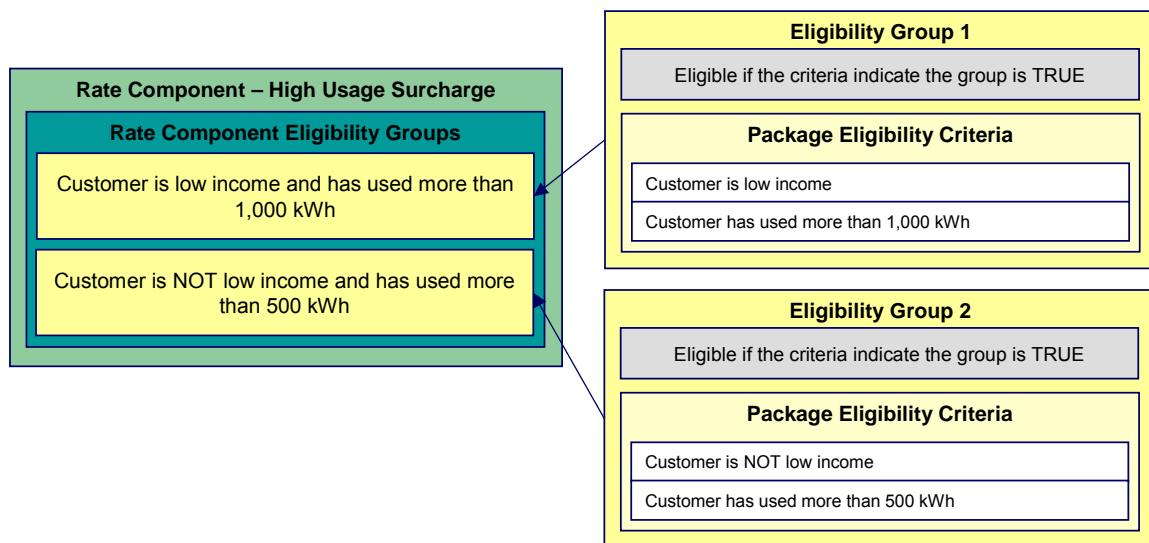
## Criteria Groups versus Eligibility Criteria

Before we provide examples of eligibility criteria, we need to explain two concepts: Criteria Groups and Eligibility Criteria. A rate component's criteria groups control whether rate application executes a rate component. At a high level, it works like this:

- A criteria group has one or more eligibility criteria. A group's criteria control whether the group is considered TRUE or FALSE.
- When you create a group, you define what should happen if the group is TRUE or FALSE. You have the following choices:
  - The rate component should be applied
  - The rate component should be skipped
  - The next group should be checked

We'll use the following example to help illustrate these points. Assume a rate component is only eligible if:

- The customer is a low income customer and has used more than 1,000 kWh
- OR, the customer isn't a low income customer and has used more than 500 kWh



This rate component requires two eligibility groups because it has two distinct conditions:

- IF (Customer is low income AND the customer has used more than 1,000 kWh)
- IF (Customer is NOT low income AND the customer has used more than 500 kWh)

If either condition is true, then the rate component is eligible.

You'd need to setup the following criteria groups in order to support this requirement:

Group No.	Group Description	If Group is TRUE	If Group is FALSE
1	Customer is a low income customer and has used more than 1,000 kWh	<i>Eligible</i>	<i>Check next group</i>
2	Customer is NOT a low income customer and has used more than 500 kWh	<i>Eligible</i>	<i>Ineligible</i>

The following criteria will be required for each of the above groups:

Group 1: Customer is a low income customer and has used more than 1,000 kWh				
Seq	Logical Criteria	If Eligibility Criteria is TRUE	If Eligibility Criteria is FALSE	If Insufficient Data
10	Customer is low income	<i>Check next condition</i>	<i>Group is false</i>	<i>Group is false</i>
20	Customer has used more than 1,000 kWh	<i>Group is true</i>	<i>Group is false</i>	<i>Group is false</i>

Group 2: Customer is NOT a low income customer and has used more than 500 kWh				
Seq	Logical Criteria	If Eligibility Criteria is TRUE	If Eligibility Criteria is FALSE	If Insufficient Data
10	Customer is NOT low income	<i>Check next condition</i>	<i>Group is false</i>	<i>Group is false</i>
20	Customer has used more than 500 kWh	<i>Group is true</i>	<i>Group is false</i>	<i>Group is false</i>

The next section describes how you'd setup the specific logical criteria in each of the groups.

## Defining Logical Criteria

When you setup an eligibility criterion, you must define two things:

- The field to be compared
- The comparison method

You have the following choices in respect of identifying the **field to be compared**:

- You can retrieve a characteristic value linked to any of the following:
  - The service agreement being billed
  - The service agreement's account
  - The main person linked to the service agreement's account
  - The characteristic premise linked to the service agreement
  - One of the service points linked to the service agreement
  - In addition, you can also use a characteristic value that is derived while the rate is being calculated (characteristic values can be created by SQ rules, register rules and many other rate component algorithms)
- You can retrieve the value of a given service quantity
- You can retrieve the final value of an earlier rate component
- You can execute an algorithm to retrieve a field value from someplace else in the system. This is a very powerful feature, but it's not terribly intuitive. We'll present a few examples later in this section to illustrate the power of this approach.

You have the following choices in respect of identifying the ***comparison method***:

- You can choose an operator (e.g., >, <, =, BETWEEN, IN, etc.) and a comparison value.
- You can execute an algorithm that performs the comparison (and returns TRUE, FALSE or INSUFFICIENT DATA). This is also a very powerful feature, but it's not terribly intuitive. We'll present a few examples later in this section to illustrate the power of this approach.

The [Examples Of Rate Component Eligibility Rules](#) provide examples to help you understand this design.

## Examples Of Rate Component Eligibility Rules

The topics in this section provide examples about how to setup rate component eligibility rules.

### Contents

- A Rate Component With A Service Quantity Comparison
- A Rate Component With A Time Span Comparison
- A Rate component With Service Type Comparison

#### A Rate Component With A Service Quantity Comparison

Imagine a rate component that is only eligible if the customer is a low-income customer and has used more than 1,000 kWh. This rate component would need the following eligibility rules:

- Customer is a low income customer
- The customer has used more than 1,000 kWh during the bill period

These rules require only one eligibility group on the rate component. It would look as follows:

Group No.	Group Description	If Group is TRUE	If Group is FALSE
1	Customer is a low income customer and has used more than 1,000 kWh	<i>Apply rate component</i>	<i>Skip rate component</i>

The following criteria will be required for this group:

Group 1: Has gas service and user is part of user group A, B or C					
Seq	Field to Compare	Comparison Method	If TRUE	If FALSE	If Insufficient Data
10	Main person characteristic: Characteristic type = Low income indicator	= YES	<i>Check next condition</i>	<i>Group is false</i>	<i>Group is false</i>
20	SQ value: UOM = KWH	>= 1,000	<i>Group is true</i>	<i>Group is false</i>	<i>Group is false</i>

Both criteria are similar – they retrieve a value and compare it to something:

The first criteria is fairly simple:

- Field to Compare. We indicate that the value of the Low Income Indicator characteristic type kept on the main person linked to the account should be retrieved. We've assumed that this characteristic type has values of **YES** and **NO**.
- Comparison Method. We indicate that if the person has a characteristic value of **YES**, then we should check the next condition; otherwise we indicate the group is false (and if the group is false, the rate component is skipped).

The second criterion works similarly:

- Field to Compare. We indicate the service quantity (SQ) that holds the total amount of KWH should be retrieved.
- Comparison Method. We simply compare the value to 1,000.

If the second criterion is true, then the group is true and the rate component will be applied; otherwise, the group is false and the rate component will be skipped.

### A Rate Component With A Time Span Comparison

A rate component that is only eligible for senior citizens has the following eligibility rule:

- Birth date equates to that of a senior citizen

This rule requires only one eligibility group on the rate component. It would look as follows:

Group No.	Group Description	If Group is TRUE	If Group is FALSE
1	Customer Is A Senior Citizen	<i>Apply rate component</i>	<i>Skip rate component</i>

The following criterion is required for this group:

Group 1: Residential, Calif, Senior					
Seq	Field to Compare	Comparison Method	If TRUE	If FALSE	If Insufficient Data
10	Main person characteristic: Date of Birth	Algorithm: True if senior	<i>Group is true</i>	<i>Group is false</i>	<i>Group is false</i>

The criterion contains a time span comparison. Time span comparisons are used to compare a date to something. In our example, we have to determine the age of the customer based on their birth date. If the resultant age is > 65, they are considered to be a senior citizen. To pull this off, you can take advantage of a comparison algorithm supplied with the base rate component as described below.

- Field to Compare. The person characteristic in which the customer's birth date is held is selected.
- Comparison Method. We chose a comparison algorithm that returns a value of **TRUE** if the related field value (the customer's date of birth) is greater than 65 years (refer to [RECC-TIMESPN](#) for an example of this type of algorithm).

You'll notice that if a value of **TRUE** is returned by the **True if senior** algorithm, the group is true (and we've setup the group to indicate a true group means the rate component is eligible).

**The time span algorithm can be used to compare days, weeks, months, etc.** Refer to [RECC-TIMESPN](#) for more information about this algorithm.

### A Rate component With Service Type Comparison

Imagine a rate component that is only eligible if the current customer has gas service AND electric service. This rate component would need the following eligibility rules:

- Customer has gas service, and
- Customer has electric service

These rules require only one eligibility group on the rate component. It would look as follows:

Group No.	Group Description	If Group is TRUE	If Group is FALSE
1	Has gas service AND electric service	<i>Apply rate component</i>	<i>Skip rate component</i>

The following criteria will be required for this group:

Group 1: Has gas service AND electric service					
Seq	Field to Compare	Comparison Method	If TRUE	If FALSE	If Insufficient Data
10	Algorithm: check if customer has gas service	= TRUE	<i>Check next condition</i>	<i>Group is false</i>	<i>Group is false</i>
20	Algorithm: check if customer has electric service	= TRUE	<i>Group is true</i>	<i>Group is false</i>	<i>Group is false</i>

Both criteria are similar – they call an algorithm that performs a logical comparison. These algorithms are a bit counter intuitive (but understanding them will provide you with another way to implement complex eligibility criteria):

Both criterion works as follows:

- Field to Compare. We chose a “field to compare” algorithm that checks if the current account has service agreements that belong to a given set of service types. It returns a value of **TRUE** if the customer has an active service agreement that matches one of the service types in the algorithm. In our example, the “check if customer has gas service” algorithm returns a value of **TRUE** if the customer has at least one active service agreement whose SA type references the gas service type. The “check if customer has electric service” algorithm is almost identical, only the service type differs. Refer to [RECF-SRVTY](#) and [RECF-AUTOPAY](#) for examples of this type of algorithm.
- Comparison Method. We simply compare the value returned by the algorithm to TRUE and indicate the appropriate response.

**Bottom line.** The “field to compare” algorithm isn’t actually returning a specific field’s value. Rather, it’s returning a value of **TRUE** or **FALSE**. This value is, in turn, compared by the “comparison method” and the group is set to true, false or check next accordingly.

# Defining Rate Components

We strongly recommend that you familiarize yourself with the information under [Designing Rate Components](#) before using this transaction.

Rate components control exactly how your bill calculation lines are calculated. Every rate component exists in respect of a given [rate version](#). This is because the rate version defines the date on which all of its rate components become effective.

**Copying, delete and moving rate components.** You can use the rate component maintenance transaction to duplicate, delete and move rate components. However, you may find these functions are easier to do using the [Rate Version Merge](#) transaction as this transaction can copy en masse and it also updates the relevant cross reference information if you move or delete a rate component.

The topics in this section describe how to maintain a rate component.

**Don't forget to finish the rate version.** After you have added all necessary rate components to a rate version, don't forget to return to [Rate Version - Main](#) and change the state of the rate version to **Finished** (otherwise, it cannot be used by billing).

## Contents

- [Rate Component - Main Information](#)
- [Rate Component - Cross Reference](#)
- [Rate Component - GL Distribution](#)
- [Rate Component - Characteristics](#)
- [Rate Component - Eligibility](#)

## Rate Component - Main Information

Core information about a rate component is defined using **Main Menu, Rates, Rate Component**.

### Description of Page

**Rate Component** contains a concatenation of basic information about the rate component. This information only appears after the rate component has been added to the database. The adjacent up and down arrows cause the rate component immediately before or after this rate component to be displayed.

**Rate Version** defines the rate version to which the rate component is linked. **Sequence** defines the relative position of this rate component in respect of the other rate components. The position is important because it defines the order in which the system calculates the charges AND the order in which the calculated charges appear on bills. **Rate Version** and **Sequence** form the unique identifier of the rate component. Both fields become protected after the rate component is added to the database.

**Re-sequencing rate components.** You must use the [Rate Version Merge](#) transaction to reposition a rate component. If you use the up / down arrows on the [Rate Version Merge](#), the system will change the rate component's **Sequence** and change all rate components that reference the old **Sequence** accordingly.

**Leave gaps in the sequence numbers.** Make sure you leave space between sequence numbers so that you can add new rate components between existing ones in the future. If you run out of space between rate components, you can use the [Rate Version Merge](#) transaction to re-sequence a rate version's rate components.

**Eligibility criteria are highlighted.** If the rate component has [eligibility criteria](#), an indication of such appears.

Select the **Rate Component Type** that corresponds with the rate component. The screen will display the appropriate fields, which correspond with the categories of rate components described in [Designing Rate Components](#). This field will be gray when the rate component is referenced on other rate components.

**Warning!** The Rate Component Type affects what you can enter on other parts of the window. The remainder of this section is devoted to those fields that can be entered regardless of RC Type. The subtopics that follow describe those fields whose entry is contingent on the RC Type.

Use **Description** to describe what the rate component does.

**Rounding Type** and **Precision** control how the system rounds the rate component's calculated value. Refer to [Rate Component Rounding](#) for a complete description.

**Default note.** **Rounding Type** defaults to **Nearest** and **Precision** defaults to a value consistent with the decimal positions defined on the rate schedule's currency code.

Turn on **For Calculation Purposes Only** (FCPO) if this rate component exists purely to calculate the percent / flat rate / unit rate used by another rate component. When this switch is on, the system does not include the calculated amount in the bill total.

When this switch is on:

- You must define what is calculated in **Result Type**; permissible values are **Unit Rate**, **Charge**, **Step Multiplier** and **Percent**.
- Use **Create a Bill Line** to define if a bill line should be created for the rate component (if you don't need to show the results to the customer or a user, there is no need to create a bill line).
- If a bill line is being created, use the **FCPO Retention Rule** to define if the FCPO amount should be set to zero on the bill line or whether the bill line's amount should be set equal to the FCPO amount.

For examples of rates that use the FCPO functionality, refer to [Calculated Minimum Charges](#), [Hours-Based Electric Rate](#) and [Step Boundaries Based On The Number Of Days Of Usage And The Region In Which The Customer Resides](#).

For each rate component, you will need to select the **Value Type** and **Value Source**. Valid values for the Value Type are **Charge**, **Percentage** and **Unit Rate**. More information to help determine which value type to choose are described in the How To sections below. Based on the Value Source entered, the remainder of this row will change.

If the Value Source is **Bill Factor**, then a prompt to indicate the **Bill Factor** will appear.

If the Value Source is **Other Rate Component**, then a prompt to indicate the **RC Sequence** will appear.

If the Value Source is **Value**, then a prompt to indicate the **Value** directly on the rate component will appear.

If the Value Source is **Value Algorithm**, then a prompt to indicate the **Value Algorithm** will appear.

If you plan to use this method, you must set up this algorithm in the system. This can be done with the following options:

- Create a new algorithm (refer to [Setting Up Algorithms](#)).
- On this algorithm, reference an Algorithm Type that calculates a rate component's value. Click [here](#) to see the algorithm types available for this plug-in spot.

For more information about which method to use, refer to [Designing Rate Components](#).

If the charges associated with the rate component are only levied during a specific season, you must turn on the **Seasonal** options.

When **Seasonal** is turned on, you use **Prorate Method** to define how the system should prorate a seasonal charge when a bill segment's start and end dates are not entirely within the seasonal period (defined in **Season**). The following options are available:

- **Accounting Date Dependent.** This option will not prorate the charge calculated by this rate component. Rather, it will only calculate the charge if the accounting date associated with the bill segment is within the seasonal period.
- **Bill End Date.** This option will not prorate the charge calculated by this rate component. Rather, it will only calculate the charge if the bill segment end date is within the seasonal period.
- **Bill Start Date.** This option will not prorate the charge calculated by this rate component. Rather, it will only calculate the charge if the bill segment [consumption start](#) date is within the seasonal period.
- **Prorate.** This option will prorate the charge calculated by this rate component based on the number of days in the adjacent season that is included in the bill period. For example, if a bill period is from 15-April through 15-May and the seasonal period is from 1-May through 31-October, the system will assume 15 / 30 of the charge should be levied.
- **Prorate seasonal SQ.** This option is only pertinent if you have:
  - Seasonal service quantity (SQ) charges, AND
  - Meters that measure seasonal consumption on separate registers, AND
  - You have a bill that crosses seasonal boundaries, AND
  - You have multiple rate versions effective across the bill period.

If this option is used, the system will prorate the charge calculated by a SQ rate component based on the number of days in the adjacent season that is included in the rate version's calculation period. For example, if two rate versions are in effect on a bill segment and one of them only becomes effective during the summer; only the summer consumption should be priced using the rate version (i.e., winter consumption should not be used on a rate version that is only effective during the summer period).

**Note.** The **Seasonal** period is defined in the two adjacent fields. The first field contains the day and month when the season starts; the second field contains the day and month when the season ends. The day and month should be entered in the format defined in your [display profile](#).

**Override Seasonal Proration.** If the seasonal functionality provided with the system does not work for your organization, you may override the logic using the **Override Seasonal Proration** plug-in spot on the [installation](#) record. For example, perhaps the seasonal period is determined dynamically based on the scheduled meter read date of the bill cycle. Or perhaps a seasonal rate component should not be prorated and the date used to determine whether the rate component is applicable is not in the list provided by the system.

**Description On Bill, Print, and Print If Zero** all control if the rate component contributes a line to the customer's bill. These fields can also be modified on [Rate Version - Bill Print Info](#). You might find it easier to setup these fields on the rate version transaction as you can copy and paste descriptions between rate components. Refer to [Rate Version - Bill Print Info](#) for a description of these fields.

The other fields on this page are dependent on the type of rate component. See the "how to" subtopics below for more information.

## How To

### Contents

- [How To Set Up Flat Charge Rate Components](#)
- [How To Set Up Service Quantity Rate Components](#)
- [How To Set Up Item Type Rate Components](#)
- [How To Set Up Apply To Rate Components](#)
- [How To Set Up Maximum Charge Rate Components](#)
- [How To Set Up Minimum Charge Rate Components](#)
- [How To Set Up Exact Charge Rate Components](#)
- [How To Set Up Summary Rate Components](#)
- [How To Set Up Interval Pricing Rate Components](#)
- [How To Set Up TOU Pricing Rate Components](#)
- [How To Set Up Calculation Algorithm Rate Components](#)

#### How To Set Up Flat Charge Rate Components

There are no special additional fields on this page available for setting up a Flat rate component. The following information will help you determine how to set up your Flat rate components.

The **Value Type** in this case would be **Charge**. This field will be gray when the rate component is referenced on another rate component.

Refer to [Rate Component - Main Information](#) for information about defining the Value Source.

Enter the verbiage to appear on the customer's bill in **Description On Bill** and turn on the **Print** switch. Refer to [How to use Description on Bill](#) for more information about these fields.

Move to the **GL Distribution** window to define how to book moneys associated with this rate component in the general ledger.

### How To Set Up Service Quantity Rate Components

When setting up an SQ rate component, additional fields become available for you to define.

The following information will help you to set up your SQ rate components.

Select a **Value Type** of **Unit Rate** if the type of charge is an amount per some unit of consumption. In the rare situation where the amount of the charge is flat for the first X units of consumption, use the Value Type of **Charge**. This field will be gray when the rate component is referenced on another rate component.

Refer to [Rate Component - Main Information](#) for information about defining the Value Source.

The amount that you specified using the bill factor / value field was a price per some unit of some thing. This thing must be identified using the **Unit Of Measure**, **Time Of Use**, and **Service Qty Identifier** fields.

For more information about UOM / TOU / SQI, refer to [UOM versus TOU versus SQI](#).

Turn on **Error if No Value** if a bill error should be generated if the UOM / TOU / SQI specified on the rate component was not supplied at billing time. For example, if every customer using a time-of-use rate should be on a meter with ON and OFF peak register, you'd turn on this switch for the ON and OFF rate components. This way, if a customer somehow was put on a time-of-use rate, but their meter wasn't a time-of-use meter, an error would be generated.

If the charge is only applicable to consumption within some tier (e.g., the first 500 therms, or all kilowatt hours in excess of 1,000), indicate that the rate component is **Stepped**. When this option is turned on, you must indicate the **Step Low Value** and **Step High Value** that the consumption charge is applicable to. For example, if the charge is applicable to the first 300 kilowatt hours, you would enter **Step Low Value** 0 and **Step High Value** 300. When multiple tiers exist, the high value of the first tier is the low value of the second tier (and so on). To extend our example, the second tier would have a low value of 300; the high value may be left blank because the system assumes a blank high value means "to infinity".

If the step boundaries are dynamic (i.e., they are calculated based on something that's only known at billing time), you can change the step boundaries using either of the following methods:

- Use **Step RC Sequence** to indicate the sequence number of the rate component whose value will be multiplied by the step's boundaries to calculate the step boundaries to use at billing time.
- Use **Step Algorithm** to indicate that an algorithm will be called to manipulate the step boundaries (note, this algorithm can manipulate the low and/or high boundaries). If you plan to use this method, 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 manipulates a rate component's step boundaries. Click [here](#) to see the algorithm types available for this plug-in spot.

**Retaining Step Algorithm High/Low Boundaries.** Step Algorithms are invoked during rate application to adjust step values prior to the system performing step value proration. Depending on the step value proration that occurs, the step value may be modified by the system after the rate component step algorithm has adjusted it. If you wish the adjusted step value to be modified by the system during step value proration, you must set up the **Adjust Rate Component Step Values After Proration** option type on the [Financial Transaction Options Feature Configuration](#). The **Adjust Rate Component Step Values After Proration** option must have a value of Y in order to have the adjusted step values modified by the system. Otherwise, set the value to N.

For an example of when dynamic steps are needed, refer to [Hours-Based Electric Rate](#) and [Step Boundaries Based On The Number Of Days Of Usage And The Region In Which The Customer Resides](#).

**Warning!** If you dynamically calculate step boundaries and the resultant high AND low values become zero, this signals the system that the rate component should be skipped (i.e., no bill line will be produced for the rate component).

Turn on **GL Statistical Quantity** if GL journal lines generated for this rate component should also contain the service quantity amount as a statistical quantity. You would use this option if you keep track of both dollar amounts and consumption units in your general ledger.

**Warning!** When a bill segment is created, the system stores the statistical quantity on the journal line associated with the rate component's distribution code. If you book multiple UOM's to the same distribution code, you should only turn on GL Statistical Qty on one of the rate components. Otherwise, you will commingle different UOM's on a journal line's statistical quantity.

Refer to [Determining the Statistical Quantity](#) for more information.

Turn on **Measures Peak Qty** if the UOM / TOU / SQI is associated with electric demand.

**Default note.** The value of **Measures Peak Qty** is defaulted from the **UOM** table (if a UOM is specified).

Enter the verbiage to appear on the customer's bill in **Description On Bill** and turn on the **Print** switch. Refer to [How to use Description on Bill](#) for more information about these fields.

Move to the **GL Distribution** window to define how to book moneys associated with this rate component in the general ledger.

#### How To Set Up Item Type Rate Components

When setting up an Item Type rate component, additional fields become available for you to define.

The following information will help you to set up your Item Type rate components.

Select a **Value Type** of **Unit Rate**. This field will be gray when the rate component is referenced on another rate component.

Refer to [Rate Component - Main Information](#) for information about defining the Value Source.

Enter the item's **Item Type**.

Turn on **Error if No Value** if a bill error should be generated if the item type specified on the rate component was not supplied at billing time. For example, if every customer using a lamp rate must have at least one 20K lumen mercury vapor lamp and one pole, you'd turn this switch for the lamp and pole rate components.

Enter the verbiage to appear on the customer's bill in **Description On Bill** and turn on the **Print** switch. Refer to [How to use Description on Bill](#) for more information about these fields.

Move to the **GL Distribution** window to define how to book moneys associated with this rate component in the general ledger.

**Multiple bill lines.** Most rate components cause one bill line to appear on a customer's bill. Item type rate components are unusual. If there are effective-dated changes to the customer's items during a bill period, one line will be generated for each time period. For example, if a customer had 2 lamps installed from 1-Jan to 15-Jan and had 4 lamps installed from 16-Jan to 31-Jan and the bill is from 1-Jan to 31-Jan, two lines will appear on the customer's bill – one for each period.

#### How To Set Up Apply To Rate Components

There are no special additional fields on this page available for setting up an Apply To rate component. The following information will help you determine how to set up your Apply To rate components.

Select a **Value Type** of **Percent** if the type of charge is a percent of the dollar value of previously calculated rate components. If the amount being charged is an amount per some unit of consumption, select the Value Type of **Unit Rate**. This field will be gray when the rate component references other rate components.

Refer to [Rate Component - Main Information](#) for information about defining the Value Source.

Enter the verbiage to appear on the customer's bill in **Description On Bill** and turn on the **Print** switch. Refer to [How to use Description on Bill](#) for more information about these fields.

Move to the **Cross Reference** window to define the rate components to which the charge will be applied.

Move to the **GL Distribution** window to define how to book moneys associated with this rate component in the general ledger.

#### How To Set Up Maximum Charge Rate Components

There are no special additional fields on this page available for setting up a Maximum rate component. The following information will help you determine how to set up your Maximum rate components.

Select a **Value Type** of **Charge**.

Refer to [Rate Component - Main Information](#) for information about defining the Value Source.

Enter the verbiage to appear on the customer's bill in **Description On Bill** and turn on the **Print** switch. Refer to [How to use Description on Bill](#) for more information about these fields.

Move to the **Cross Reference** page to define the rate components against which the maximum charge will be applied.

Move to the **GL Distribution** page to define how any additional revenue associated with the maximum charge should be booked in the general ledger.

#### How To Set Up Minimum Charge Rate Components

There are no special additional fields on this page available for setting up a Minimum rate component. The following information will help you determine how to set up your Minimum rate components.

Select a **Value Type** of **Charge**.

Refer to [Rate Component - Main Information](#) for information about defining the Value Source.

Enter the verbiage to appear on the customer's bill in **Description On Bill** and turn on the **Print** switch. Refer to [How to use Description on Bill](#) for more information about these fields.

Move to the **Cross Reference** window to define the rate components against which the minimum charge will be applied.

Move to the **GL Distribution** window to define how any additional revenue associated with the minimum charge should be booked in the general ledger.

#### How To Set Up Exact Charge Rate Components

There are no special additional fields on this page available for setting up an Exact Charge rate component. The following information will help you determine how to set up your Exact Charge rate components.

Select a **Value Type** of **Charge**.

Refer to [Rate Component - Main Information](#) for information about defining the Value Source.

Enter the verbiage to appear on the customer's bill in **Description On Bill** and turn on the **Print** switch. Refer to [How to use Description on Bill](#) for more information about these fields.

Move to the **Cross Reference** window to define the rate components against which the exact charge will be applied.

Move to the **GL Distribution** window to define how any additional revenue associated with the minimum charge should be booked in the general ledger.

#### How To Set Up Summary Rate Components

Summary rate components use no other fields on this page other than those defined above.

Enter the verbiage to appear on the customer's bill in **Description On Bill** and turn on the **Print** switch. Refer to [How to use Description on Bill](#) for more information about these fields.

Move to the **Cross Reference** window to define the rate components to summarize on this rate component.

#### How To Set Up Interval Pricing Rate Components

For more information about setting up interval pricing rate components, refer to [Setting Up Interval Pricing Rate Components](#) in the Interval Billing documentation.

#### How To Set Up TOU Pricing Rate Components

For more information about setting up TOU pricing rate components, refer to [Setting Up TOU Pricing Rate Components](#) in the Interval Billing documentation.

## How To Set Up Calculation Algorithm Rate Components

Calculation Algorithm rate components are used when none of the other rate components will provide you with the functionality you need.

First, you must select a **Calc Algorithm**. This calculation algorithm will have the responsibility of producing your bill lines. All rate component options become available when you use this rate component type, based on the assumption that it is generally capable of doing anything you might require.

If you plan to use this method, you must set up this algorithm in the system. This can be done with the following options:

- Create a new algorithm (refer to [Setting Up Algorithms](#)).
- On this algorithm, reference an Algorithm Type that calculates a rate component. Click [here](#) to see the algorithm types available for this plug-in spot.

**Note.** The calculation algorithm's main purpose is to create bill calculation lines. However, the algorithm may populate other information for the bill, for example, it may add to the SQ or register read collection or it may overwrite the description on bill.

If you wish to provide your users with the ability to audit the bill lines created by a rate component of this type, you may indicate an **Audit Algorithm**. Currently the only types of audit algorithms supplied by the system are those which [audit interval pricing details](#) or those which [audit TOU pricing details](#).

Refer to [Setting Up Interval Pricing Rate Components](#) and [Setting Up TOU Pricing Rate Components](#) for more information about the audit algorithms provided with the system.

All the other fields on the page are optional fields available to you for use by your calculation algorithm.

Turn on **Error if No Value** if a bill error should be generated if the UOM / TOU / SQL specified on the rate component was not supplied at billing time.

Turn on the **Derive SQ** switch when your calculation algorithm will be creating entries in the SQ array for the UOM / TOU / SQL defined on your rate component. In this case, the system will not attempt to locate an entry in the SQ collection matching the rate component's UOM / TOU / SQL before calling your calculation algorithm.

Enter the verbiage to appear on the customer's bill in **Description On Bill** and turn on the **Print** switch. Refer to [How to use Description on Bill](#) for more information about these fields.

Move to the **GL Distribution** window to define how any additional revenue associated with the minimum charge should be booked in the general ledger.

## Rate Component - Cross Reference

Minimum charge, maximum charge, exact charge, apply-to, and summary rate components reference other rate components. For these types of rate components, open **Main Menu, Rates, Rate Component** and navigate to the **Cross Reference** tab define the other rate components to which it refers.

### Description of Page

**Rate Component** contains a concatenation of basic information about the rate component. This information only appears after the rate component has been added to the database. The adjacent up and down arrows cause the rate component immediately before or after this rate component to be displayed.

Use the **Apply this Rate Component to** collection to specify the other rate component(s) on the Rate Version to which the rate component in question should refer. The list of rate components available for reference is displayed on the right. Use the left arrow to move the desired rate components into the grid. If a rate component appears in the apply to list in error, use the left arrow to remove the rate component.

Refer to [Editable Grid](#) in the system wide standards documentation for more information about adding records to a collection by selecting from a list.

**Note.** An apply to rate component that has a charge type of **unit rate** can only reference SQ, item, interval pricing and calculation rate components. Apply to rate components with a charge type of **percentage**, along with summary, minimum charge, maximum charge and exact charge rate components can reference all types of rate components including summary.

The **This RC is referenced by** collection indicates what other rate components reference the rate component in question.

## Rate Component - GL Distribution

Open **Main Menu, Rates, Rate Component** and navigate to the **GL Distribution** tab to define each rate component's effect on the general ledger. GL distribution information is required for all rate components except for FCPO and summary rate components.

### Description of Page

**Rate Component** contains a concatenation of basic information about the rate component. This information only appears after the rate component has been added to the database. The adjacent up and down arrows cause the rate component immediately before or after this rate component to be displayed.

If the rate component's charges should be booked to a single GL account regardless of the service agreement's SA Type's Revenue Class, enter a **Distribution Code**. If the rate component's charges are booked to a different GL account depending on the service agreement's Revenue Class, turn on **Use Revenue Class** and insert the appropriate **Revenue Class** and **Distribution Code** for each that uses the rate.

## Rate Component - Characteristics

Your algorithms and reports may need additional rate component fields that aren't supported in the base-package. If so, you can set up a rate component characteristic for each such field.

**Snapshotting characteristics on bill calculation lines.** The system will automatically copy a rate component's characteristics onto the resultant bill lines for those characteristic types that are usable on bill lines.

Open **Main Menu, Rates, Rate Component** and navigate to the **Characteristics** tab to define characteristic values for your rate component.

#### Description of Page

For each characteristic in the collection, indicate the appropriate **Characteristic Type** and **Characteristic Value**.

**Note.** You can only choose characteristic types defined as permissible on the rate schedule record. Refer to [Setting Up Characteristic Types & Their Values](#) for more information.

## Rate Component - Eligibility

This page is used to define the conditions under which a rate component will be applied. If you don't specify any **Eligibility Criteria Groups**, the system assumes the rate component should be applied (i.e., a rate component is eligible by default).

Refer to [The Big Picture Of Rate Component Eligibility](#) for more information.

Open this page using **Main Menu, Rates, Rate Component** and navigate to the **Eligibility** tab.

#### Description of Page

**Rate Component** contains a concatenation of basic information about the rate component. This information only appears after the rate component has been added to the database. The adjacent up and down arrows cause the rate component immediately before or after this rate component to be displayed.

**Warning!** The following information is not intuitive; we strongly recommend that you follow the guidelines under [The Big Picture Of Rate Component Eligibility](#) before attempting to enter this information.

The **Eligibility Criteria Group** scroll contains one entry for each group of eligibility criteria. The following fields may be defined for each group:

- Use **Sequence** to control the relative order in which the group is executed when the system determines if the rate component should be applied (smaller numbers are executed before larger numbers).
- Use **Description** and **Long Description** to describe the criteria group.
- Use **If Group is True** to define what should happen if the eligibility criteria (defined in the following grid) return a value of **True**.
  - Choose **Apply Rate Component** if this rate component should be executed.
  - Choose **Check Next Group** if the next criteria group should be checked.
  - Choose **Skip Rate Component** if this rate component should NOT be executed.
- Use **If Group is False** to define what should happen if the eligibility criteria (defined in the following grid) return a value of **False**.
  - Choose **Apply Rate Component** if this rate component should be executed.

- Choose **Check Next Group** if the next criteria group should be checked.
- Choose **Skip Rate Component** if this rate component should NOT be executed.

The grid that follows contains the rate component's eligibility criteria. Think of each row as an "if statement" that can result in the related eligibility group being True or False. For example, you might have a row that indicates the customer is eligible for the rate component if the customer's birth date equates to that of a senior citizen. The following bullets provide a brief description of each field on an eligibility criterion. Please refer to [Defining Logical Criteria](#) for several examples of how this information can be used.

- Use **Sequence** to control the order in which the criteria are checked.
- Use **Criteria Field** to define the field to compare:
  - Choose **Characteristic** if you want to compare a characteristic value that resides on any of the following to a given value (use the adjacent fields to define the object on which the characteristic resides and the characteristic type):
    - The service agreement being billed
    - The service agreement's account
    - The main person linked to the service agreement's account
    - The characteristic premise linked to the service agreement
    - One of the service points linked to the service agreement (selected at random)
    - The bill segment's bill
    - In addition, you can also use a characteristic value that is derived while the rate is being calculated (characteristic values can be created by SQ rules, register rules and many other rate component algorithms)

**Note.** For effective-dated characteristics, the system selects the characteristic value effective on the start date of the bill segment consumption period. Refer to [Determine The Consumption Period For Metered Service](#) for information about how the consumption period is calculated before applying the rate.

- Get the most effective characteristic value for the given characteristic type and bill segment start date
- Choose **Rate Component Value** if you want to compare the final value of an earlier rate component to a given value. Push the adjacent search button to select the rate component's sequence number.
- Choose **Service Quantity** if you want to compare the final value of a service quantity to a given value. Push the adjacent search button to select the service quantity's **UOM / TOU / SQI**.
- Choose **Algorithm** if you want to compare anything other than the above options. Push the adjacent search button to select the algorithm that is responsible for retrieving the comparison value.
- Use **Criteria Comparison** to define the method of comparison:
  - Choose **Algorithm** if you want an algorithm to perform the comparison and return a value of True, False or Insufficient Data.

- Choose any other option if you want to compare the **Criteria Field** using a logical operator. The following options are available:
  - Use **>**, **<**, **=**, **>=**, **<=**, **<>** (not equal) to compare the **Criteria Field** using standard logical operators. Enter the comparison value in the adjacent field.
  - Use **In** to compare the **Criteria Field** to a list of values. Each value is separated by a comma. For example, if a field value must equal **1**, **3** or **9**, you would enter a comparison value of **1,3,9**.
  - Use **Between** to compare the **Criteria Field** to a range of values. For example, if a field value must be between **1** and **9**, you would enter a comparison value of **1,9**. Note, the comparison is inclusive of the low and high values.

**Note.** The comparison value field expects the comma as a delimiter between values. This means that if the values involve decimals, the comma cannot be used as a decimal symbol regardless of what your implementation has set up in your [display profile](#). For decimal values, use **.** as the decimal symbol.

- The next three fields control whether the related logical criteria cause the eligibility group to be considered True or False:
  - Use **If True** to control what happens if the related logical criterion returns a value of True. You have the options of **Group is true**, **Group is false**, or **Check next condition**. If you indicate **Group is true** or **Group is false**, then the rate component will be **Skipped** or **Applied** based on the values defined above in **If Group is False** and **If Group is True**.
  - Use **If False** to control what happens if the related logical criterion returns a value of False. You have the options of **Group is true**, **Group is false**, or **Check next condition**. If you indicate **Group is true** or **Group is false**, then the rate component will be **Skipped** or **Applied** based on the values defined above in **If Group is False** and **If Group is True**.
  - Use **If Insufficient Data** to control what happens if the related logical criterion returns a value of "Insufficient Data". You have the options of **Group is true**, **Group is false**, or **Check next condition**. If you indicate **Group is true** or **Group is false**, then the rate component will be **Skipped** or **Applied** based on the values defined above in **If Group is False** and **If Group is True**.

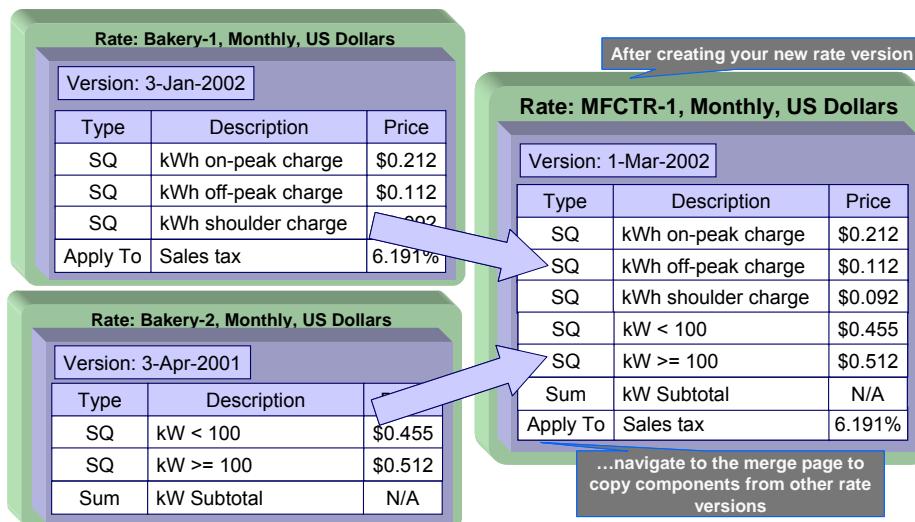
## Rate Version Merge

The following point summarize the many diverse functions available on this transaction:

- A rate version's rate components have unique sequence numbers. This number controls the order in which a rate component is processed. If you need to insert a rate component and there's no space between existing sequence numbers, you can use this transaction to renumber the rate version's rate components.
- You can use this transaction to move a rate component to a different position within a rate version. When a rate version is moved, all references to the rate component are changed to reflect the new sequence number.
- You can use this transaction to delete a rate component. When a rate component is deleted, all references to the rate component are deleted.

- You can use this transaction to copy rate components from other rate versions. For example,
  - You may want to create a rate version that is similar to an existing rate version. Rather than copying all the information from the existing rate version and then removing the inapplicable components, this page may be used to selectively copy specific rate components from the existing rate version to the new rate version.
  - You may have customer-specific rates that are very similar, but still unique. You can use this transaction to build a customer-specific rate. In this scenario, you may choose to create special ‘mini’ rate versions, one for each of the various options. Then, you could use the rate version merge page to select the components applicable for the new custom rate version.

**The target rate version must exist prior to using this page.** If you are creating a new rate, you must first create the [Rate Schedule](#) and [Rate Version](#) and then navigate to the merge page to copy rate component information.



**Duplicate versus Merge.** The [Rate Version](#) page itself has [duplication](#) capability. You would duplicate a rate version if you want to a) create a new rate version AND b) populate it with all the rate components from an existing rate version. The [Rate Component Maintenance](#) transaction also allows you to [duplicate](#) a rate component within a rate version.

Open **Main Menu, Rates, Rate Version Merge** to open this page.

### Description of Page

Select the **Original Rate Version**, which is the target for merging the rate component information.

**Finished rate versions are protected.** Only rate versions that are **In Progress** or **Validated** may be used as the target rate version in the merge functionality. If any changes are made to a **Validated** rate version on this page, the system will move the status back to **In Progress**. If a rate version with a status of **Finished** is selected, it will be displayed, but you will be unable to copy any information to it (and its description will appear in red). If you need to make changes to a **Finished** rate version, navigate to [Rate Version – Main](#) and change the rate version's status to **In Progress**. Refer to [Lifecycle of a Rate Version](#) for more information.

Select the **Merge From Rate Version**, which is your template rate version to copy the rate components from.

**Note.** You may only copy components from one Merge From rate version at a time. If you want to copy components from more than one rate version, select the first Merge From rate version, copy the desired records, Save, and then select the next Merge From rate version.

The left portion of the page displays any existing rate components for the **Original Rate Version**. The right portion of the page displays the existing rate components for the **Merge From Rate Version**.

You can use the **Copy All** button to copy all the rate components from the **Merge From** rate to the **Original** rate. If you use **Copy All**, please be aware that the rate components are added to the end of the **Original** rate version.

Each time you save the changes, the system renames the **Original** rate version using the **Start From Sequence Number** and **Increment By**.

The left portion of the page initially displays existing rate component records linked to the original rate version. In the **Merge Type**, you will see the word **Original**, for any of these records. The **Sequence**, **RC Type** and **Description** for each rate component are displayed. In the right portion of the page, the existing records in the merge from rate version are displayed initially.

The topics that follow describe how to perform common maintenance tasks:

## Contents

- [Resequencing Rate Components](#)
- [Removing A Row From A Grid](#)
- [Adding A New Row To A Rate](#)
- [Removing An Uncommitted Row From A Rate](#)
- [Moving Positional Rows Up and Down](#)
- [Rate Component Cross Reference Information](#)

## Resequencing Rate Components

If there are no gaps between two rate components and you need to resequence the rate components:

- Make a change to the **Start From Sequence Number** or **Increment By** (so that the Save button becomes enabled).
- Click the Save button.

## Removing A Row From A Grid

If you wish to remove a record linked to the Original rate version, press the “-“ button to the left of the record.

## Adding A New Row To A Rate

You may move any of the rate components from the Merge From rate version to the original rate version 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 Rate

If you have copied a row across by mistake, you may remove it by clicking on the right arrow adjacent to the appropriate record.

## Moving Positional Rows Up and Down

A rate component's sequence number controls the order in which the rate component should be applied. You may modify the order execution using the up and down arrows.

The sequences need to be renumbered in order to ensure the correct order. The fields **Start from sequence number** and **Increment By** are used to define the sequence number to assign to the first rate component and the subsequent values to assign to subsequent rate components.

Refer to [Editable Grid](#) in the system wide standards documentation for more information about adding records to a collection by selecting from a list and repositioning rows within a grid.

## Rate Component Cross Reference Information

All information related to a rate component, including general ledger information, characteristics and eligibility rules, will be copied.

Special logic exists for merged rate components that reference other rate components. This includes the cross-reference collection and the eligibility criteria.

This logic is difficult to explain succinctly, so we'll use an example. Imagine that the Merge From rate components look as follows:

	Seq	RC Type	Description
◆	10	Service Quantity	0 to 300 kWh step
◆	20	Service Quantity	300 to max kWh
◆	30	Summary	Summary of stepped charges
◆	32	Service Quantity	Days Calculation Rate
◆	34	Minimum Charge	Minimum Charge
◆	100	Apply-to	Employee discount (rider req)
◆	200	Apply-to	City tax
◆	210	Apply-to	State tax

The [Rate Component - Cross Reference](#) tab for the merge from rate version shows the following cross-reference information for sequence **210**:

Apply This Rate Component To:		
	RC Sequence	Description
➡️➡️	10	0 to 300 kWh step
➡️➡️	20	300 to max kWh
➡️➡️	34	Minimum Charge
➡️➡️	100	Employee discount (rider req)

Assume that only rate components 10, 20 and 210 are copied to the new rate version and assume that the Original rate version already contains a rate component with sequence number 10 that is moved to the second position.

Start from sequence number		10	Increment By	10
	Merge Type	Sequence	RC Type	Description
➡️➡️	Merge	10	Service Quantity	0 to 300 kWh step
➡️➡️	Original	10	Flat Charge	Customer Service Charge
➡️➡️	Merge	20	Service Quantity	300 to max kWh
➡️➡️	Merge	210	Apply-to	State tax

After saving, the second service quantity rate component is renumbered and the State tax rate component will reflect the renumbering in the cross-reference and will not include the rate components that were not copied.

Here is the view on the merge page after saving:

Start from sequence number		10	Increment By	10
	Merge Type	Sequence	RC Type	Description
➡️➡️	Original	10	Service Quantity	0 to 300 kWh step
➡️➡️	Original	20	Flat Charge	Customer Service Charge
➡️➡️	Original	30	Service Quantity	300 to max kWh
➡️➡️	Original	40	Apply-to	State tax

If you navigate to the [Rate Component - Cross Reference](#) tab for the original rate version for sequence **40 – State tax**, it will look as follows:

Apply This Rate Component To:		
	Sequence	Description
➡️➡️	10	0 to 300 kWh step
➡️➡️	30	300 to max kWh

If any rate component refers to another rate component in the eligibility criteria, the sequence number of the eligibility criteria is also adjusted accordingly.

**Note.** You will be warned by the system when you copy a rate component that has cross-references to other rate components that are not being copied.

## Rate Check

The Rate Check page allows you to test your rates under various scenarios. The topics in this section describe how to use this transaction.

### Contents

- [Rate Check - Main](#)
- [Rate Check Results - Calc Lines](#)
- [Rate Check Results - SQ Details](#)
- [Rate Check Results - Register Read](#)
- [Rate Check Results - Characteristics](#)

### Rate Check - Main

Open **Main Menu, Rates, Rate Check** to enter the parameters required to check a rate.

#### Description of Page

**Easiest way to use rate check.** The easiest way to populate the above parameters is to select an existing bill segment on [Bill Segment – Calc Lines](#) and then use the rate schedule context menu to navigate to this transaction. When you do this, the system populates the above fields from the values on the bill segment. After these fields are populated, you can override the defaulted values to check various scenarios.

Indicate the primary **Rate Schedule** that you want to test. The version(s) used will be those in effect during the start and end dates. Only **validated** and **finished** rate versions will be used.

**Default note.** If **Rate Schedule** is blank and you enter a **SA ID**, the service agreement's current rate will be defaulted.

While using a **SA ID** is not strictly required to check a rate, we strongly recommend finding a representative service agreement to check a rate. Why? Because charges in a rate may be affected by values in a service agreement. For example, consider the following types of rate components that use service agreement-specific values:

- Rate components with bill factors that are subject to the presence of contract riders. In fact, if a bill factor requires a contract rider and you do not specify a **SA ID** rate check assumes that the rider is applicable.
- Rate components with bill factors that are subject to tax exemptions.
- Rate components with bill factors with customer-specific prices.
- Rate components with service quantity (SQ) rules that use characteristics on service agreement, account or the service agreement's characteristic premise.
- Rate components with SQ rules that use contract quantities.

In other words, if you don't specify a **SA ID**, you will not be able to check rates with the above characteristics.

Enter a **Bill Seg ID** if you want to test the rate with actual usage for a given bill segment. This is useful if you want to compare the actual bill segment results with the results of the same usage under a different rate schedule. If you indicate a **Bill Seg ID**, rate check's **SA ID** will be populated with the bill segment's service agreement ID (and **SA ID** will be protected).

**Overriding consumption.** If you enter a **Bill Seg ID**, the bill segment's consumption will be defaulted into the consumption array that follows. You may override this consumption to test various scenarios.

You must indicate a **Revenue Class**. This is needed for applying the appropriate GL distribution code for rate components where the GL account is based on the customer's revenue class.

Enter the **Start Date** and **End Date** that you are testing. If a Bill Seg ID is entered, the start and end dates are taken from the bill segment's period, but can be overwritten.

**Consumption Period.** Refer to [Determine The Consumption Period For Metered Service](#) for information about how the consumption period is calculated before applying the rate.

Indicate the **Accounting Date**. If a **Bill Seg ID** is entered, the accounting date on the bill segment is used but can be overwritten.

Enter the quantities that you want to test in the quantity collection. For every **UOM/TOD/SQI** combination, enter a register quantity.

**Default note.** If you indicate a bill segment, the quantity grid will be populated with data from the bill segment's meter read snapshot collection. This includes the **UOM/TOD/SQI** quantities and the **Service Points** associated with these quantities.

Enter the **Service Point ID** if your rate uses register rules that require characteristic values from the service point or premise. For example, the register rule that converts CCF to therms uses a bill factor to store the conversion factor. This bill factor may have been configured to use characteristics on the service point / premise (e.g., the conversion factor may differ based on the storage location associated with the service point).

**Overriding service points.** If you specify a **SA ID**, you may indicate a service point that is not connected to your service agreement (if you want to experiment with the effect of different service point characteristics on the rate).

The collection of **Characteristic Types** and **Characteristic Values** is used to apply the appropriate bill factor values. If these values are entered, these values will be used to apply the appropriate bill factors (even if you have specified a service agreement from which characteristics can be derived). If these values are blank AND there is no SA ID, then you will receive an error from the rate application indicating which values are needed.

Indicate the **Bill Language** to use for the resulting bill calc line descriptions.

When you have entered everything, click on the **Check Rate** button. This will cause your rate to be applied to the parameters entered above. If the system is successful, you will be transferred to the **Calc Lines** page on which the resulting calculation details can be viewed.

## Rate Check Results - Calc Lines

After you have pressed the **Check Rate** button on the **Main** page, you will be transferred to the **Calc Lines** tab to view the lines that have been calculated by your rate.

### Description of Page

The **Rate Schedule** and its description, the **Revenue Class** and its description and the **Bill Period** are displayed at the top.

More than one Rate Version may have been effective for the primary Rate Schedule (and any applicable sub rates) during this bill period. If so, multiple collections of calculation lines will be produced. Use the **Bill Seg Hdr** scroll to view each collection of calculation lines.

For each Bill Seg Hdr, the following information will be displayed above the calculation line grid.

<b>Sequence</b>	This identifies which Bill Seg Hdr is being displayed.
<b>Start Date, End Date</b>	This is the period, within the bill period, that the rate version, which produced these lines, was effective.
<b>Amount</b>	This is the total amount calculated by this group of calculation lines.
<b>Desc on Bill</b>	The description on bill for the rate version that produced these lines.
<b>Rate Version</b>	The rate version that was used to produce these lines.

The grid, which displays the bill calculation lines for this rate version, shows one row exists for every calculation involved in this process. This information is very similar to what is displayed on the bill calculation lines page for a bill segment. The following information is displayed in the grid:

- **Sequence** is the system-assigned unique identifier of the calculation detail row.
- **Description on Bill** is the information about the bill line that appears on the customer's bill.
- **Calculated Amount** is the calculated amount associated with the bill line.
- The **Print** switch controls whether information about this line will print on the customer's bill.
- The **App in Summary** switch defines if this line's amount also appears on a summary line. This switch is turned on if the corresponding rate component is summarized on a summary rate component.
- **UOM** is the unit of measure of the service quantity priced on the calculation line.
- **TOU** is the time-of-use code of the service quantity priced on the calculation line.
- **SQI** is the service quantity identifier of the service quantity priced on the calculation line.
- **Billable Service Quantity** is the service quantity priced on the calculation line. This quantity will be different from the measured consumption if there are SQ rules or register rules in effect.
- **Base Amount** is used by calculation lines (e.g. taxes) that are cross-referenced to other calculation lines and whose value(s), therefore, depend on the amounts calculated by those other lines. The Base Amount shows the total amount derived from the cross-referenced line(s) that the current line then used to calculate its billed amount.

- **Sequence** refers to the sequence number of the rate component on the applicable rate version that was used to calculate the line.
- **Meas Peak Qty** is checked if the UOM priced on the calculation line is used to measure a peak quantity.
- **Exempt Amount** is the amount of the calculated charge that the customer doesn't have to pay because they are tax exempt. This will only be calculated if you have provided an SA ID on the main page.
- **Distribution Code** is the distribution code associated with the calculation line. This distribution code is used to build the general ledger details on the bill segment's financial transaction.

## Rate Check Results - SQ Details

After you have pressed the **Check Rate** button on the **Main** page, you can navigate to the **SQ Details** tab to view the service quantities that supplied to (or derived by) your rate. Please be aware that a rate can derive service quantities if there are register rules, service quantity rules or time-of-use mapping rate components.

### Description of Page

The **Rate Schedule** and its description, the **Revenue Class** and its description and the **Bill Period** are displayed at the top.

The SQ details grid is a snapshot of the service quantities amassed from:

- The quantities input on the main page OR the quantities taken from the service quantity collection for the bill segment, if it was input on the main page.
- The SQ rules defined on the input rate.
- The register rules defined on the input rate.

The following information is displayed in the grid:

- **UOM** is the unit of measure of the service quantity.
- **TOU** is the time-of-use code of the service quantity.
- **SQI** is the service quantity identifier of the service quantity.
- **Initial Service Quantity** is the initial quantity amassed by the system before application of the rate's SQ rules.
- **Billable Service Quantity** is the service quantity that will be priced by the service agreement's rate. This amount differs from the initial service quantity when the rate's SQ rules manipulate consumption.

## Rate Check Results - Register Read

After you have pressed the **Check Rate** button on the **Main** page, you can navigate to the **Register Read** tab to view the final quantities (Final UOM, Final TOU and Final SQI) that were derived by your rate's register rules.

### Description of Page

The **Rate Schedule** and its description, the **Revenue Class** and its description and the **Bill Period** are displayed at the top.

This page displays measured quantities from one of two places:

- From the values input on the main page of Rate Check
- From the Register Read collection for the bill segment indicated on the main page of Rate Check

The final quantities will reflect any changes to the measured quantities resulting from applied register rules.

The following information is displayed in the grid:

- **UOM** is the unit of measure of the measured quantity.
- **TOU** is the time-of-use code of the measured quantity.
- **SQI** is the service quantity identifier of the measured quantity.
- **Measured Quantity** is the initial quantity amassed by the system before application of the rate's register rules.
- **Final UOM** is the unit of measure of the resultant billing quantity.
- **Final TOU** is the time-of-use code of the resultant billing quantity.
- **Final SQI** is the service quantity identifier of the resultant billing quantity.
- **Register Quantity** is the resultant billing quantity.
- **Service Point ID** displays information about the service point used by rate check to determine any characteristic values needed for the register rules.

## Rate Check Results - Characteristics

After you have pressed the **Check Rate** button on the **Main** page, you can navigate to the **Characteristics** tab to view the final characteristics collection as manipulated by your rate's plug-ins.

### Description of Page

The **Rate Schedule** and its description, the **Revenue Class** and its description and the **Bill Period** are displayed at the top.

The collection of **Characteristic Types** and **Characteristic Values** reflects any changes made to the initial characteristics collection (input on the main tab page) after applying the rate.

## Use Sub Rate Schedules To Encapsulate Common Rate Components

Many implementations have multiple rate schedules that share common rate components. For example, the following illustration shows two rate schedules that use identical rate components to calculate taxes:

**Two rate schedules with identical rate components that are used to levy taxes**

**Rate Schedule ERES1 (Stepped rate)**

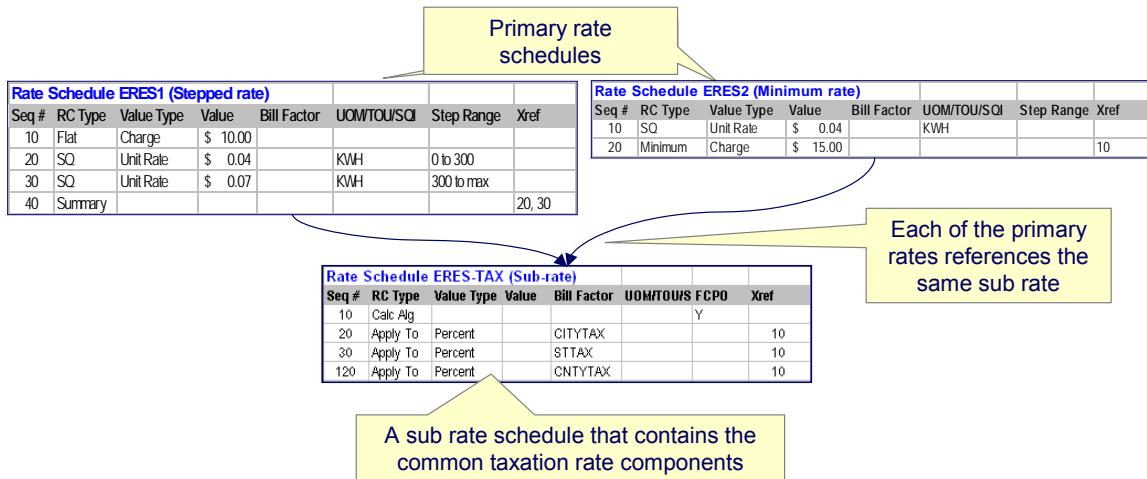
Seq #	RC Type	Value Type	Value	Bill Factor	UOM/TOU/SQI	Step Range	Xref
10	Flat	Charge	\$10.00				
20	SQ	Unit Rate	\$ 0.04		KWH	0 to 300	
30	SQ	Unit Rate	\$ 0.07		KWH	300 to max	
40	Summary						20, 30
50	Apply To	Percent		CITYTAX			10, 20, 30
60	Apply To	Percent		STTAX			10, 20, 30
70	Apply To	Percent		CNTYTAX			10, 20, 30

**Rate Schedule ERES2 (Minimum rate)**

Seq #	RC Type	Value Type	Value	Bill Factor	UOM/TOU/SQI	Step Range	Xref
10	SQ	Unit Rate	\$ 0.04		KWH		
20	Minimum	Charge	\$15.00				10
30	Apply To	Percent		CITYTAX			10, 20
40	Apply To	Percent		STTAX			10, 20
50	Apply To	Percent		CNTYTAX			10, 20

### Two Rate Schedules That Share Common Rate Components

Rather than duplicate the taxation rate components on both rates, a sub rate schedule can be constructed that references the shared rate components. The sub rate schedule can then be included on each of the primary rate schedules:



### Setting Up A Sub Rate Schedule To Encapsulate Common Rate Components

The primary reason to encapsulate common rate components into sub rate schedules is to simplify long-term maintenance. For example, if a new tax is introduced, only the sub rate schedule in the above example would require modification. The converse is also true. For example, if pricing structures are changed, only the affected primary rate schedules require modification. Both of these scenarios result in reduced rate construction and testing time.

The topics in this section describe sub rate schedules in more detail.

### Contents

- Sub Rate Schedules and Primary Rate Schedules Have Similar Features
- Primary Rates May Reference Many Sub Rates
- How To Set Up A Sub Rate Schedule

## Sub Rate Schedules and Primary Rate Schedules Have Similar Features

Most of the rich functionality described above is shared by both sub rate schedules and primary rate schedules. The following points highlight the major differences:

- Only primary rate schedules can be referenced on a customer's service agreement and on an adjustment type's generation algorithm.
- Only primary rate schedules can reference sub rate schedules.
- Sub rate schedules cannot reference a service type, a currency code, the allow estimates indicator, register rules or valid meter configurations. This is because these attributes are defined for the primary rate schedule used to drive the calculation.
- When charges are calculated for a primary rate schedule's sub rate schedules, the primary rate schedule's [frequency](#) is used. This allows a single sub rate schedule to be referenced on any primary rate schedule regardless of the primary rate schedule's frequency. This means that sub rate schedules do not reference a frequency.
- Because sub rate schedules inherit the frequency of the primary rate schedule at execution time, [Rate Check](#) cannot be used to verify sub rate schedules directly. Rather, use a primary rate schedule that references the sub rate schedule on Rate Check.

## Primary Rates May Reference Many Sub Rates

There is no limit to the number of sub rate schedules that can be referenced on a primary rate schedule. If multiple sub rate schedules are defined on a primary rate schedule, the primary rate schedule defines the invocation order of each.

A bill segment that is calculated using a rate will contain separate bill calculation details for each primary rate schedule and sub rate schedule used in its calculation. Please be aware of the following special scenarios that can affect the number of bill calculation details on a bill segment:

- If a given primary or sub rate schedule has multiple rate versions effective during a bill segment's calculation period, a separate bill calculation detail will be created for each version of every primary and sub rate schedule.
- It is possible for a sub rate schedule to not contribute any charges to a bill segment (for example, if its rate components have eligibility rules that inhibit the creation of calculation lines). If a sub rate schedule does not contribute additional charges or service quantities to a bill segment, the system will suppress the bill calculation detail. In other words, an empty bill calculation detail is not created if a subsidiary rate schedule contributes nothing new to a bill segment.

An adjustment that is calculated using a primary rate schedule will contain a separate calculation line for every rate component on the primary and sub rate schedules that contribute charges to the adjustment.

## How To Set Up A Sub Rate Schedule

Sub rate schedules are set up using the same transactions as primary rate schedules. With the exception of the differences outlined above, the design and construction of sub rate schedules is virtually identical to that of primary rate schedules.

The major difference occurs when a sub rate schedule has rate components that require the value of lines calculated by the primary rate schedule or earlier sub rate schedules. Rate components that levy taxes using the monetary value calculated by earlier rate schedules are good examples. In order to implement these types of rate components on a sub rate schedule, follow these instructions:

- Set up a **Rate Component - Calculation Algorithm** that references the base-package algorithm type called **C1-RC-CLAMT**. Algorithms of this type calculate the value of a rate component by summarizing the value of lines calculated by earlier rate components. Please note that you can limit the summarized lines to those calculated by rate components that are marked with a given characteristic type and value.
- Add a rate component to the sub rate schedule:
  - Choose a **Rate Component Type** of **Calculation Algorithm**.
  - Select the **Calculation Algorithm** defined above.
  - Turn on **FCPO** (for calculation purposes only) to prevent the value of this line from being added to the total charges.
- Set up **Apply To** rate components as normal for each tax to be calculated. These rate components will reference the rate component described above in their cross reference.

## **Setting Up Estimated Consumption For Items**

Some types of physical devices (i.e., items) use power but a meter does NOT measure the power consumed. The most common example is a street light. If this situation exists for your company AND if the amount charged for the lamp varies based on the amount of estimated power used, then you will have to maintain estimated DAILY consumption for each type of item. The system uses this information at billing time to estimate how much power was used by the customer's items and then prices the estimated consumption as per the terms in the service quantity rate component.

Select **Admin Menu, Item Type SQ Estimate** to define estimated consumption.

### **Description of Page**

Select the **Item Type** using the search.

Enter the estimated daily consumption in the **Daily Service Quantity**. Each Service Quantity has an **Effective Date** so that changes in the estimated quantities may be recorded over time.

**Note.** The unit of measure of the daily service quantity is displayed adjacent to the item type description. This value is defined on the item type table. Refer to [Setting Up Item Types](#) for more information.

### **Where Used**

This information is used when the system calculates a bill segment for a service agreement that references service points with items. If the items reference item types for which the system should estimate consumption, the system extracts the daily consumption amount from this table.

## Setting Up Degree Days (and Average Temperatures)

Many organizations print degree-day and/or average temperature information on their customers' bills. Select **Admin Menu, Degree Days** to define these values for a given trend area.

### Description of Page

The easiest way to define degree days is to select a **Trend Area** and **Month In Year** and tab out of the field. The system generates one row in the **Degree Days** collection for each day of the month. After the days exist in the collection, you should define the following for each day:

<b>Degree Day</b>	This is a unit of measure used to express the extent to which temperatures vary from a specific reference temperature on a given date.
<b>Maximum Temperature</b>	This is the maximum temperature on a given date.
<b>Minimum Temperature</b>	This is the minimum temperature on a given date.
<b>Average Temperature</b>	This is the average temperature on a given date.

### Where Used

This information may be used by your bill print routines.

## Effective Dates & Price Proration

In this section, we describe how proration applies to different types of rate components. Proration implies assessing charges proportionately.

The term "proration" describes two different issues:

- Prorating a charge whose value changes during a bill period. For example, if a tax rate changes during a bill period and you've indicated that such changes should be prorated, the system prorates the tax change (e.g., 20 days at 5% and 10 days at 6%).
- Prorating charges when the time period being billed is not in sync with the time period in which the charges were defined in the rate. For example, if a rate contains a flat monthly charge and the bill period spans two months, the flat charge must be prorated.

This is a complicated topic as it's possible for many proration issues to exist on a single bill. For example, on a single bill:

- The rate structure can change several times during the bill period (i.e., multiple rate versions are effective).
- The taxing authority changes the tax rate.
- The customer's tax exemption changes.
- The customer-specific price can change.
- etc.

### Contents

[Types of Proration](#)  
[Proration Factors](#)  
[Overriding Proration Factors](#)  
[Service Quantity Proration](#)  
[Rate Component Value Proration](#)  
[Bill Factor Value Proration](#)  
[Example of Generating Prorated Bill Calculation Lines](#)  
[Seasonal Proration](#)  
[Price Changes and Interval Quantities](#)

## Types of Proration

The following section describes the types of proration performed by applying a rate.

For more information on how prorated charges are presented on the printed bill, refer to [How To Use Description On Bill](#).

### Contents

[Consumption Period Proration](#)  
[Calculation Period Proration](#)  
[Bill Factor Proration](#)  
[A Service Agreement Rate Changes Are Not Prorated](#)

### Consumption Period Proration

A rate schedule specifies its frequency. Bill segments generated using a rate are expected to have a consumption period that corresponds to this frequency. When a bill segment's consumption period falls outside of the rate frequency's normal period, proratable service quantities may be adjusted to match the normal period for calculation. The service quantities are then re-adjusted based on calculation period proration.

Refer to [Service Quantity Proration](#) for more information about these calculations.

### Calculation Period Proration

There can be multiple versions of a given rate over time. Each version has an effective date. When the system detects that a rate schedule has multiple versions of a rate in effect during a bill period, it may or may not prorate the charges. You control exactly how the system handles this situation using the rate schedule's proration parameters.

If multiple rate versions are effective during the consumption period and the rate schedule is configured to prorate the rate versions, a given bill segment will contain a set of bill calculation lines for each rate version. You should consider quite carefully how to handle this situation on your printed bills as rate version proration can confuse a customer.

Refer to [Defining Rate Versions](#) for more information about rate versions.

## Bill Factor Proration

A rate contains rate components. The primary reason a rate component exists is to define how much to charge for the services sold by your company. When you create rate components, you have two ways to indicate how much to charge:

- You can specify the value directly in a rate component.
- You can have the rate component use a bill factor that contains the value.

Bill factors are effective-dated (i.e., their values may change over time). When the system detects that a rate component's bill factor has multiple values in effect during a bill period, it may or may not prorate the charges. You control exactly how the system handles this situation using the bill factor's proration parameters.

Refer to [Defining Bill Factors](#) and [Bill Factor Value Proration](#) for more information.

### Contents

- [Tax Exemption Proration](#)
- [Contract Value Proration](#)
- [Contract Rider Proration](#)

#### Tax Exemption Proration

Bill factors that levy taxes may or may not be eligible for tax exemptions; it's up to the respective taxing authority. Customers who are eligible for tax exemptions will have their tax obligation reduced (or nullified) at billing time.

Whether or not a customer is tax exempt (and the percent exempt) is defined on the service agreement's tax exemption page. A customer's tax exemptability is effective-dated (i.e., it may change over time). If a customer's tax exemptability changes during a bill period, the system may or may not prorate the charges. You control exactly how the system handles this situation using the bill factor's proration parameters.

#### Contract Value Proration

Some types of bill factors do not have the same value for all customers. For example, you may have rates where the price per kilowatt-hour can be different for every customer using the rate.

For bill factors whose value is customer-specific, the value is defined on the service agreement's contract value page. The contract value is effective-dated (i.e., it may change over time). If a customer's contract value changes during a bill period, the system may or may not prorate the changes. You control exactly how the system handles this situation using the bill factor's proration parameters.

#### Contract Rider Proration

Contract riders are used when the charge associated with a bill factor is only applicable to some customers for some period of time. For example, if a bill factor is used to provide a discount during the first 3 months of service, then a contract rider would be used.

A customer's contract rider applicability period is defined on the service agreement's contract rider page. A rider's applicability may change during a customer's bill period. You control exactly what the system does when a rider's applicability changes during a bill period using the bill factor's proration parameters.

## A Service Agreement Rate Changes Are Not Prorated

A service agreement's rate is effective-dated (i.e., it may change over time). When the system calculates a service agreement's bill segment when multiple rates are effective during the bill period, it will use a single rate from the service agreement's rate history. The system chooses the rate as per the service agreement's SA type's rate selection date.

Refer to [SA Type - Rate](#) for more information.

## Proration Factors

The purpose of rate components is to generate bill calculation lines. When rate application builds bill calculation lines, it may also prorate the charges.

There are multiple types of proration performed by rate application. Service quantities, step boundaries, rate values, and bill factor values may all be prorated as part of applying a rate.

We need to establish some terms in order to understand the calculations.

### Contents

- [Normal Days](#)
- [Consumption Period Factor](#)
- [Calculation Period Factor](#)
- [BF Value Period Factor](#)

### Normal Days

As mentioned previously, the rate schedule references a frequency that specifies the number of times per year that service agreements referencing the rate are expected to bill. The following formula dictates a rate frequency's normal days: 365 days / number of periods per year.

Example:

- For a monthly frequency (12 periods per year),  $365 / 12 = 30$  days.
- For a quarterly frequency (4 periods per year),  $365 / 4 = 91$  days.

Refer to [Defining Frequency Codes](#) for more information.

### Consumption Period Factor

Before applying a rate, the bill segment's consumption period is determined. The ratio of normal days to consumption period days is the consumption period factor.

The system uses the Minimum and Maximum Days Offset on the rate's [Frequency](#) to determine whether or not the rate component should be prorated due to a short or long consumption period.

If the number of days in the consumption period is within the tolerance of days determined by the specified on the rate's frequency, the consumption period factor is 1 (not applicable).

Example:

If the consumption period spans 60 days and the rate's normal days equals 30 days then the consumption period factor is:

- $30 \text{ days} / 60 \text{ days} = 0.5.$

If the consumption period spans 31 days and the rate's normal days equals 30 days and the maximum offset is 3 days, the consumption period factor is 1 (not applicable) because the consumption period days falls within the tolerance specified on the frequency.

For more information on how a bill segment's consumption period is determined, refer to [How The System Determines How Much Was Consumed](#). For more information on specifying minimum and maximum offset days on frequency, refer to [Defining Frequency Codes](#).

### Calculation Period Factor

The calculation period is the date range where a rate version is effective during the consumption period. For each rate version that is effective during the consumption period, a bill segment calculation period is created. A set of bill calculation lines is created for each calculation period.

A calculation period factor is determined for each calculation period. The calculation period factor is the ratio of calculation period days to normal days. If the consumption period factor is 1 (not applicable), then the calculation period factor is the ratio of calculation period days to consumption period days.

Example:

If the consumption period covers April 1 to April 30 and there was a new effective rate version as of April 21, there are two calculation periods.

- Calculation period April 1 to April 20, calculation period factor =  $20 \text{ days} / 30 \text{ days} = 0.6666667.$
- Calculation period April 21 to April 30, calculation period factor =  $10 \text{ days} / 30 \text{ days} = 0.3333333.$

### BF Value Period Factor

The bill factor value period is the date range where a proratable bill factor value is effective during the calculation period.

When a bill factor value changes during the calculation period, the results is that multiple bill calculation lines are generated for the same rate component.

The ratio of the number of days within the calculation period that a distinct value is effective to the number of days in the calculation period determines the BF value period factor. When the bill factor is set up not to prorate, the BF value period factor is 1 (not applicable).

Example:

If the state tax rate changes from 6% to 6.5% on April 16 and the calculation period is April 1 to April 30, there are two bill factor value periods.

- Bill factor value period April 1 to April 15, BF value period factor =  $15 \text{ days} / 30 \text{ days} = 0.5.$
- Bill factor value period April 16 to April 30, BF value period factor =  $15 \text{ days} / 30 \text{ days} = 0.5.$

Refer to [Defining Bill Factors](#) for more information on specifying whether the bill factor allows proration.

## Overriding Proration Factors

If the system logic for determining proration factors does not satisfy your requirements, you may plug in an override proration algorithm to calculate the proration factors as required by your implementation. The override proration algorithm must be specified on the [Installation](#) record.

The system is delivered with an algorithm type that sets the proration factors to 1 (not applicable) when a predefined characteristic is specified on a rate component. Refer to [OVPF-NOPROR](#) for more information.

## Service Quantity Proration

When a bill calculation line is generated for a service quantity type rate component, the applicable service quantity is multiplied by the consumption period factor. That product is then multiplied by the calculation period factor to determine the prorated service quantity for the bill calculation line.

Example:

Customer's electricity bill segment's consumption period is from April 1 through May 15. A new rate version for the rate schedule is effective as of April 16. 100 kWh is consumed.

Assuming a monthly frequency, if the consumption period spans 45 days and the first calculation period spans 15 days, and the applicable service quantity is 100 kWh; then:

- Consumption period factor = 30 normal days / 45 days = 0.6666667.
- Calculation period factor = 15 days / 30 normal days = 0.5.
- Prorated service quantity =  $100 \text{ kWh} \times 0.6666667 \times 0.5 = 33 \text{ kWh}$ .

The next calculation period extends to the end of the consumption period (30 days):

- Consumption period factor does not change = 0.6666667.
- Calculation period factor = 30 days / 30 days = 1.0.
- Prorated service quantity =  $100 \text{ kWh} \times 0.6666667 \times 1.0 = 67 \text{ kWh}$ .

### Contents

- [Measures Peak](#)
- [Step Proration](#)
- [Step Multipliers](#)

### Measures Peak

If the rate component specifies that the service quantity measures peak, then service quantity proration is not performed. Although service quantities are not prorated if a rate component measures peak, rate component values (like price per kW) are prorated.

Refer to [Rate Component Value Proration](#) for more information about how rate component values are prorated.

## Step Proration

If a rate component's charges are based on stepped pricing then the step boundaries are also prorated (as long as the rate component is not marked as measures peak). Step boundaries are multiplied by the calculation period factor to determine the adjusted steps boundaries.

Example:

Assuming a monthly frequency, if the rate component's step boundaries are 100 kWh to 300 kWh, and the calculation period is 45 days; the step boundaries are prorated as:

- 45 days / 30 days x 100 kWh = 150 kWh.
- 45 days / 30 days x 300 kWh = 450 kWh.

## Step Multipliers

If the rate component specifies that the steps be multiplied by the result value of another rate component, step proration is not performed. It is assumed that the result value from the referenced rate component has already been prorated, as the referenced step rate component is subject to [Rate Component Value Proration](#).

## Rate Component Value Proration

Except for summary type rate components (which are not prorated at all), a value is specified on the rate component. The value may be a unit rate, percentage or charge. The value may be specified directly on the rate component, derived from a bill factor, derived from another rate component or calculated by an algorithm.

Regardless of where the rate component value comes from, it will be multiplied by the calculation period factor when determining the charge on a bill calculation line. Exceptions to this rule are:

- Unit rate values where the rate component's measures peak attribute is not checked.
- Percentage values.
- Rate component values derived from another rate component.

Even though the rate component values are not prorated for unit rate and percentage value types, if these rate values change during the calculation period due to a bill factor value change, bill factor value proration still applies.

Values derived from another rate component are not subject to rate value proration, as rate component value proration was already performed on the referenced rate component. Again, if the bill factor value changed on the referenced rate component, bill factor value proration still applies.

Example:

A rate component a rate version charges \$2.00 per kW. The rate schedule's normal days equals 30, and the calculation period spans 15 days. The peak kW service quantity is 50 kW.

- The calculation period factor is 0.5 (15 days / 30 days).
- Rate component value proration yields  $\$2.00 \times 0.5 = \$1.00$  per kW.

For more information on bill factor value proration, refer to [Bill Factor Value Proration](#).

## Bill Factor Value Proration

Bill factor value proration behaves the same way any time a bill factor value changes within a calculation period. This could be due to a change in a tax rate, a contract rider becoming effective, a tax exemption expiring, etc.

Multiple bill calculation lines are generated for a given rate component when a proratable bill factor value changes during the calculation period.

Example:

If the state tax rate changes from 6% to 6.5% on April 16, and the calculation period is April 1 to April 30.

- The first BF value factor period April 1 to April 15, so BF value period factor is 0.5.
- The value used in the first bill calculation line for state tax is  $6\% \times 0.5 = 3\%$ .
- The second bill factor value period is April 16 to April 30, so the BF value period factor is 0.5.
- The value used in the second bill calculation line for state tax is  $6.5\% \times 0.5 = 3.25\%$ .

## Example of Generating Prorated Bill Calculation Lines

Let's say that 1200 kWh was consumed from March 17 to April 30. The peak kW for the consumption period is 20 kW.

A monthly rate schedule **E-COM1** exists with two rate versions.

Rate version effective prior to April 1 has the following components:

- \$0.05 per kWh.
- \$1.00 per kW.

Rate version effective as of April 1 has the following rate components:

- \$0.04 per kWh; as of April 16, \$0.07 per kWh (new effective bill factor value).
- \$1.50 per kW; as of April 16, \$1.25 per kW (new effective bill factor value).

### Calculation Period March 17 to March 31

The consumption period spans 45 days, so the consumption period factor is 30 normal days / 45 days = 0.6666667.

This calculation period spans 15 days, so the calculation period factor is 0.5.

Service quantity proration for the first calculation period yields:

- $1200 \text{ kWh} \times 0.6666667 \text{ (consumption period factor)} \times 0.5 \text{ (calculation period factor)} = 400 \text{ kWh}$ .
- $20 \text{ kW} \times 1 \text{ (not applicable)} = 20 \text{ kW}$ .

Rate component value proration for the first calculation period yields:

- $\$0.05 \times 1 \text{ (not applicable)} = \$0.05 \text{ per kWh}$ .
- $\$1.00 \times 0.5 \text{ (calculation period factor)} = \$0.50 \text{ per kW}$ .

### First Set of Bill Calculation Lines

Bill calculation line 1:  $400 \text{ kWh} \times \$0.05 = \$20.00$ .

Bill calculation line 2:  $20 \text{ kW} \times \$0.50 = \$10.00$ .

### Calculation Period April 1 to April 30

This calculation period spans 30 days, so the calculation period factor is 1.

Service quantity proration for the calculation period yields:

- $1200 \text{ kWh} \times 0.6666667 \text{ (consumption period factor)} \times 1 \text{ (calculation period factor)} = 800 \text{ kWh}$ .
- $20 \text{ kW} \times 1 \text{ (not applicable)} = 20 \text{ kW}$ .

Rate component value proration for the first calculation period yields:

- $\$0.04 \times 1 \text{ (not applicable)} \times 0.5 \text{ (BF value period factor)} = \$0.02 \text{ per kWh}$ .
- $\$0.07 \times 1 \text{ (not applicable)} \times 0.5 \text{ (BF value period factor)} = \$0.035 \text{ per kWh}$ .
- $\$1.50 \times 1 \text{ (calculation period factor)} \times 0.5 \text{ (BF value period factor)} = \$0.75 \text{ per kW}$ .
- $\$1.25 \times 1 \text{ (calculation period factor)} \times 0.5 \text{ (BF value period factor)} = \$0.625 \text{ per kW}$ .

### Second Set of Bill Calculation Lines

Bill calculation line 1:  $800 \text{ kWh} \times \$0.02 \text{ per kWh} = \$16.00$ .

Bill calculation line 2:  $800 \text{ kWh} \times \$0.035 \text{ per kWh} = \$26.00$ .

Bill calculation line 3:  $20 \text{ kW} \times \$0.75 \text{ per kW} = \$15.00$ .

Bill calculation line 4:  $20 \text{ kW} \times \$0.625 \text{ per kW} = \$12.50$ .

## Seasonal Proration

Seasonal proration adds another layer of complexity. If rate components are specified to prorate seasonally, the seasonal period specified on a given seasonal rate component dictates how a rate component's value or associated service quantity is prorated.

There are two proration methods, **Prorate** and **Prorate seasonal SQ**, which may be specified on seasonal rate components. Both of these proration methods behave the same way with respect to rate component value proration. If a given seasonal rate component is eligible for [Rate Component Value Proration](#), the ratio of days in the season that are applicable during the calculation period to the days in the calculation period determines the seasonal factor. When it comes to service quantity proration, the two proration methods are different. If a given seasonal rate component is eligible for [Service Quantity Proration](#), it prorates the service quantities as follows:

- **Prorate**. A seasonal factor is calculated as the ratio of days in the season that are applicable during the calculation period to the days in the calculation period. This ratio is applied along with the consumption period factor and calculation period factor to determine the prorated service quantity.

- **Prorate seasonal SQ.** In this case, it is assumed that either the meter measures seasonal consumption on separate registers or the consumption has been split into separate service quantities by other means (like an SQ rule). To determine a prorated seasonal service quantity, the seasonal proration factor is calculated as for a **Prorate** seasonal rate component. The seasonal factor is then multiplied by another ratio of total consumption period days to days in the season that are applicable for the total consumption period. This modified seasonal factor is applied along with the consumption period factor and calculation period factor to determine the prorated service quantity.

**Prorate seasonal SQ.** Specify this option only if you can determine the seasonal consumption as an independent service quantity for use by rate application. Because the service quantities themselves dictate the applicability of the rate component, a given service quantity need not appear in rate versions effective outside the season (i.e. a rate version effective on October 1 may not include rate components that charge for kWh/summer).

**Override Seasonal Proration.** If the seasonal functionality provided with the system does not work for your organization, you may override the logic using the **Override Seasonal Proration** plug-in spot on the [Installation](#) record. For example, perhaps the seasonal period is determined dynamically based on the scheduled meter read date of the bill cycle.

#### Example (**Prorate**):

Let's say that 600 kWh and 50 kW are the billable service quantities for a consumption period from April 1 to April 30. **Prorate** seasonal rate components are defined on a monthly rate that charge \$0.05 per kWh and \$0.75 per kW from January 1 to April 15. **Prorate** seasonal rate components are defined on the same rate version that charge \$0.06 per kWh and \$0.80 per kW between April 16 and December 31. Assume only one rate version is effective for the consumption period.

- The consumption period factor is 1 (not applicable).
- The calculation period factor is 1.0.
- The seasonal factor for the kWh rate component whose season is effective from January 1 to April 15 is 0.5 (15 season days in calculation period / 30 days in the calculation period).
- The resulting bill calculation line uses service quantity proration:
  - $600 \text{ kWh} \times 0.5 \text{ (seasonal factor)} \times \$0.05 = \$15.00$ .
- The seasonal factor for the kW rate component whose season is effective from January 1 to April 15 is also 0.5.
- The resulting bill calculation line uses rate component value proration:
  - $50 \text{ kW} \times \$0.75 \times 0.5 \text{ (seasonal factor)} = \$18.75$ .
- The seasonal factor for the kWh rate component whose season is effective from April 16 to December 31 is 0.5.
- The resulting bill calculation line uses service quantity proration:
  - $600 \text{ kWh} \times 0.5 \text{ (seasonal factor)} \times \$0.06 = \$18.00$ .
- The seasonal factor for the kW rate component whose season is effective from April 16 to December 31 is also 0.5.

- The resulting bill calculation line uses rate component value proration:
  - $50 \text{ kW} \times \$0.80 \times 0.5 \text{ (seasonal factor)} = \$20.00$ .

Example (**Prorate seasonal SQ**):

Let's say that a given meter measures winter consumption separate from summer consumption. The meter itself measures kWh/summer on a register between June 21 and September 18, and measure kWh/winter on another register at other times of the year.

We are generating a bill segment with a consumption period from September 2 to October 30. A monthly rate schedule has two rate versions effective during the consumption period. The latest rate version is effective on October 1.

Each of the rate versions has **Prorate seasonal SQ** seasonal rate components defined. The first rate version's rate components charge \$0.06 per kWh/summer and \$0.05 per kWh/winter. The second rate version has a single rate component that charges \$0.05 per kWh/winter.

Assume that the billable service quantities are 800 kWh/summer and 1600 kWh/winter.

First set of bill calculation lines (September 2 to September 30):

- The consumption period factor is 30 normal days / 60 consumption days = 0.5.
- The calculation period factor is 30 calculation days / 30 normal days = 1.0.
- Rate Component 1 (\$0.06 per kWh/summer):
  - The seasonal factor 17 seasonal days / 30 calculation days = 0.5666667.
  - The modified seasonal factor is  $0.5666667 \times 60 \text{ consumption days} / 17 \text{ seasonal days} = 2.0$ .
  - The resulting bill calculation line:
    - $800 \text{ kWh/summer} \times 0.5 \text{ (consumption period factor)} \times 2.0 \text{ (modified seasonal factor)} \times \$0.06 = \$48.00$ .
- Rate Component 2 (\$0.05 per kWh/winter):
  - The seasonal factor is 12 seasonal days / 30 calculation days = 0.4.
  - The modified seasonal factor is  $0.4 \times 60 \text{ consumption days} / 42 \text{ seasonal days} = 0.571428$ .
  - The resulting bill calculation line:
    - $1600 \text{ kWh/winter} \times 0.5 \text{ (consumption period factor)} \times 0.571428 \text{ (modified seasonal factor)} \times \$0.05 = \$22.86$ .

Second set of bill calculation lines (October 1 to October 30):

- The consumption period factor is 30 normal days / 60 consumption days = 0.5.
- The calculation period factor is 30 calculation days / 30 normal days = 1.0.
- Rate Component 1 (\$0.05 per kWh/winter):
  - The seasonal factor 30 seasonal days / 30 calculation days = 1.0.
  - The modified seasonal factor is  $1.0 \times 60 \text{ consumption days} / 42 \text{ seasonal days} = 1.428571$ .
  - The resulting bill calculation line:

- 1600 kWh/winter x 0.5 (consumption period factor) x 1.428571 (modified seasonal factor) x \$0.05 = \$57.14.

For more information on specifying seasonal proration options, refer to [Defining Rate Components](#).

## Price Changes and Interval Quantities

All of the above sections described situations where the service quantities will be prorated due to changes in the price during the billing period. The above sections assume that the bill contains a simple metering scenario, where reads will not exist on the date of the price break.

However, consider an interval billing example. There are two basic pricing scenarios for interval quantities:

- Interval prices are applied to interval quantities. In this case, proration is not applicable because the prices are changing at small intervals and the appropriate algorithms understand how to apply the prices to the interval quantities. Refer to [Interval Pricing](#) for more information.
- Time of use prices. In this case, interval quantities are being mapped to time of use codes before applying time of use prices. The standard rate component algorithm will compare interval quantities to a time of use map resulting in new entries in the bill segment's SQ collection. The algorithm will then apply the appropriate price for each SQ based on the TOU Pricing bill factor. In the time of use pricing scenario, it is possible for the TOU prices to change during the bill period. When this happens, proration should not be necessary because readings will exist for the date of the price break. Refer to [Time of Use Mapping and Pricing](#) for more information.

## Rate Examples

---

In this section, we provide examples of how to set up some charges that occur in many organizations. While your rates may not possess these exact charges, this section should help solidify your understanding of how to set up your company's rates.

### Contents

- [A Simple Electric Rate](#)
- [Adding Measured Quantities Together](#)
- [CCF To Therm Conversion](#)
- [How To Adjust Measured Consumption By A Minimum Contract Quantity](#)
- [Elevation Charges](#)
- [Gas Pressure Factor Adjustments](#)
- [Pricing Interval Consumption Using Interval Prices](#)
- [Pricing Non Interval Consumption Using Weighted Average Prices](#)
- [Employee Discounts](#)
- [Customer Specific Prices](#)
- [Hours-Based Electric Rate](#)
- [Step Boundaries Based On The Number Of Days Of Usage And The Region In Which The Customer](#)
- [Calculated Minimum Charges](#)
- [Stepped Taxes](#)

## A Simple Electric Rate

The following table illustrates the rate components necessary to levy a simple rate that contains the following charges:

- Monthly service charge: \$10.00.
- First 300 kilowatt hours at \$0.0382 per kWh.
- Remaining kilowatt hours at \$0.0673 per kWh.
- State Tax at the prevailing rate (this tax is a percentage of a dollar value).
- County Use Tax at the prevailing rate (this tax is an amount per kilowatt hour).

Seq #	RC Type	Value Type	Value	Bill Factor	Item Type	UOM/ TOU/ SQI	Step Range	Cross Reference
10	Flat Charge	Charge	10.00					
20	Service Quantity	Unit Rate	0.0382			KWH	0 to 300	
30	Service Quantity	Unit Rate	0.0673			KWH	300 to 99999999	
40	Summary							20, 30
50	Apply To	Percent		STTAX				10, 20, 30
60	Apply To	Unit Rate		COUNTY TAX				20, 30

The following table illustrates how you might want to define the Description on Bill for these rate components.

Seq #	Description on Bill
10	Monthly service charge of %R %D
20	First %Q kWh at %I per kWh %F
30	Remaining %Q kWh at %I per kWh %F
40	Subtotal
50	State sales tax %R %T
60	County use tax %R per kWh %T

## Adding Measured Quantities Together

Refer to the how to section under [Setting Up Service Quantity \(SQ\) Rules](#) for more information.

## CCF To Therm Conversion

Refer to the how to section under [Setting Up Register Rules](#) for more information.

## How To Adjust Measured Consumption By A Minimum Contract Quantity

Refer to the how to section under [Setting Up Service Quantity \(SQ\) Rules](#) for more information.

## Elevation Charges

Assume you have rates that charge a different amount per gallon depending on the elevation. For example, customers on rate X who live between sea level and 500 feet are charged \$0.001 per gallon; customers living over 500 feet are charged \$0.0015 per gallon. Customers on rate Y who live between sea level and 500 feet are charged \$0.002 per gallon; over 500 feet are charged \$0.0025.

### Contents

- [Elevation Charges - Characteristics](#)
- [Elevation Charges - Bill Factors](#)
- [Elevation Charges - Rate Schedule and Rate Components](#)

### Elevation Charges - Characteristics

Given the elevation gradients are the same for rates X and Y, you can set up a characteristic called **ELEV** and assign it characteristic values of **< 500** and **> 500**. If possible, you should indicate the appropriate characteristic value for each postal default so that new premises will receive their appropriate characteristic value. If premises already exist, you must populate this characteristic value accordingly.

### Elevation Charges - Bill Factors

Because each rate charges a different price per gallon depending on the characteristic, you'll need two bill factors – **one for rate X, the other for rate Y. Both will reference the new ELEV characteristic.** The only difference will be their bill factor values.

### Elevation Charges - Rate Schedule and Rate Components

You will create two rate schedules – one for rate X, the other for rate Y. Rate X will have a rate component that references the rate X elevation bill factor. Rate Y will have a rate component that references the rate Y elevation bill factor.

**Alternatives.** The design of the system would allow you to create a single rate rather than two rates. How? In the single rate you would have two rate components referencing the two elevation bill factors. You would indicate that both bill factors are contingent on the presence of a contract rider by turning on the bill factors' contract rider flags. When you set up a service agreement for a customer, you would create a contract rider referencing the necessary bill factor. We don't recommend this approach because you are not accurately reflecting your actual rates with corresponding rates in the system. We showed this approach to provide an alternative way of thinking about your rates.

## Gas Pressure Factor Adjustments

Assume you have rates that require a cubic foot of gas converted based on the pressure factor associated with the premise. Assume there's a different pressure factor based on the take point from which the gas is pumped.

### Contents

- [Pressure Factor - Characteristics](#)
- [Pressure Factor - Bill Factors](#)
- [Pressure Factor - Register Rule](#)
- [Pressure Factor - Rate Schedule](#)

### Pressure Factor - Characteristics

Create a characteristic called **TAKEPT** and assign it characteristic values for each take point in your service area. If possible, you should indicate the appropriate characteristic value for each postal default so that new premises will receive their appropriate characteristic value. If premises already exist, you must populate this characteristic value accordingly.

### Pressure Factor - Bill Factors

Set up a bill factor called **PF ADJ**. On the bill factor, indicate that it uses the **TAKEPT** characteristic. Create a bill factor value for each take point to define its respective pressure factor.

### Pressure Factor - Register Rule

Create a register rule called **CCFPFADJ**. Populate it with the following information:

Field	Value
Register Algorithm	<i>Bill factor based UOM conversion</i>
UOM (Source)	<i>CCF</i>
UOM (Result)	<i>CCF PF</i>
Bill Factor	<i>PF ADJ</i>

This register rule will convert a metered amount of cubic feet to a pressure factor adjusted amount of cubic feet using the bill factor.

### Pressure Factor - Rate Schedule

Link the **CCFPFADJ** register rule to every rate that requires a pressure factor adjustment.

## Pricing Interval Consumption Using Interval Prices

Assume you have interval meters that measure how much was consumed at various times during a day. You may need to price each interval read using the real time price that's effective on the read's start date and time.

**Note**, this example assumes that you do not have the interval billing subsystem and are recording your interval reads in the Meter Read table.

### Contents

- Real Time Pricing - Bill Factors
- Real Time Pricing - Service Quantity Identifier
- Real Time Pricing - Register Rule
- Interval Billing - Rate Schedule

### Real Time Pricing - Bill Factors

Set up a bill factor called **RTP**. On the bill factor, indicate that it uses the **N/A** characteristic. Also, set the bill factor type to **Interval**. You will also have to [Maintain Interval Values](#) for the bill factor.

### Real Time Pricing - Service Quantity Identifier

Set up a service quantity identifier called **\$**.

### Real Time Pricing - Register Rule

Create a register rule called **REALTIME**. Populate it with the following information:

Field	Value
Register Algorithm	<i>Real time pricing</i>
UOM	<i>KWH</i>
Bill Factor	<i>RTP</i>
SQI (Result)	<i>\$</i>

This register rule will convert a metered amount of KWH into a dollar value using the real time price in effect on the read's start date and time.

### Interval Billing - Rate Schedule

Link the **REALTIME** register rule to every rate that requires such. These rate will also require a SQ rate component with a price of \$1 and a SQI of **\$**.

## Pricing Non Interval Consumption Using Weighted Average Prices

Refer to [Interval Values Can Be Used For More Than Just Interval Meters](#) to understand the business situation.

**Contents**

- [Weighted Average Pricing - Bill Factors](#)
- [Weighted Average Pricing - RC Algorithm](#)
- [Weighted Average Pricing - Rate Component](#)

**Weighted Average Pricing - Bill Factors**

Set up a bill factor called **WAP**. If you assume you have different WAP's depending on the customer's load profile, indicate that it uses the **LOADPROF** characteristic type. Also set the bill factor type to **Interval**. You will also have to [Define Interval Values](#) for the bill factor.

**Weighted Average Pricing - RC Algorithm**

Create an algorithm called **WAP**. Populate it with the following information:

Field	Value
Algorithm Type	<a href="#">PX VALUE</a>
Bill Factor	<a href="#">RAP</a>

Refer to [PX VALUE](#) for information about this algorithm.

**Weighted Average Pricing - Rate Component**

Create a SQ rate component on the respective rate(s).

Seq #	RC Type	Value Type	Value Source	Value Algorithm
20	Service Quantity	Unit Rate	Algorithm	WAP

**Employee Discounts**

Assume your organization provides discounts of 25% to employees for their residential consumption.

**Contents**

- [Employee Discounts - Characteristics](#)
- [Employee Discounts - Bill Factors](#)
- [Employee Discounts - Rate Schedule and Rate Components](#)
- [Employee Discounts - Start Option](#)
- [Employee Discounts - Service Agreements](#)

**Employee Discounts - Characteristics**

Because the discount amount is the same for all employees, you just need a dummy characteristic to keep the relational model consistent. You should set up a characteristic type called **N/A** and with a characteristic value of **N/A**. It's important that the characteristic value is identical to the characteristic type otherwise billing will generate an error.

## Employee Discounts - Bill Factors

Set up a bill factor called **EMPDISC**. On the bill factor, indicate that it uses the characteristic described above and that it is contingent on the presence of a contract rider. Create a bill factor value with an amount of -25%.

## Employee Discounts - Rate Schedule and Rate Components

On every residential rate that is eligible for employee discounts, you will have a rate component that references the **EMPDISC** bill factor.

## Employee Discounts - Start Option

Create a start option for every SA type that can use one of your employee discount rates. The start option should contain a contract rider that references the **EMPDISC** bill factor.

## Employee Discounts - Service Agreements

On existing service agreements of employees eligible for a discount, add a contract rider referencing the **EMPDISC** bill factor.

# Customer Specific Prices

Assume your organization has some rates where the customer may negotiate specific prices.

## Contents

- [Customer Specific Prices - Bill Factors](#)
- [Customer Specific Prices - Rate Schedule and Rate Components](#)
- [Customer Specific Prices - Start Option](#)
- [Customer Specific Prices - Service Agreements](#)

## Customer Specific Prices - Bill Factors

Set up a bill factor called **KWHCUST**. On the bill factor indicate that it:

- Uses a N/A characteristic (you need a characteristic on all bill factors). Refer to [Employee Discounts](#) for an example.
- Has a value type of **Unit Rate**.
- Has its value specified in a contract term.
- Should generate an error if it finds no value in the contract.

Define the standard bill factor value to use for those customers who do not elect to negotiate a specific price.

## Customer Specific Prices - Rate Schedule and Rate Components

On every rate that may allow a customer specific price, you will have a rate component that references the **KWHCUST** bill factor.

## Customer Specific Prices - Start Option

Create a start option for every permissible price on every SA type with such rates. The start option should contain a contract value that references the **KWHCUST** bill factor. You will also have a start option to use for those customers to do not elect to negotiate a specific price. This start option will not define a contract value.

## Customer Specific Prices - Service Agreements

For existing service agreements using this rate where the customer has negotiated a specific price, add a contract value referencing the **KWHCUST** bill factor.

## Hours-Based Electric Rate

**Warning!** If your organization does not sell electric service (or if you do and don't use hour-based steps) you might want to skip this example as it involves understanding a fairly complex electric concept – kwh / kw = hours.

The following table illustrates the rate components necessary to levy a rate that contains the following charges:

- Monthly service charge: \$10.00.
- First 10 hours of use at \$0.0673 per kWh.
- Remaining hours of use at \$0.032 per kWh.
- State Tax at the prevailing rate (this tax is a percentage of a dollar value).
- County Use Tax at the prevailing rate (this tax is an amount per kilowatt hour).

Seq #	RC Type	Value Type	Value	Bill Factor	Item Type	UOM/ TOU/ SQI	Step Range	Cross Reference	Calc Only	Result Type
10	Flat Charge	Charge	10.00						N	
20	Service Quantity	Unit Rate	1.0			KW			Y	Unit Rate
30	Service Quantity	Unit Rate	0.0673			KWH	0 to 10 Multiply by RC: 20		N	
40	Service Quantity	Unit Rate	0.032			KWH	10 to 9999999 Multiply by RC: 20		N	
50	Summary							30, 40	N	
60	Apply To	Percent		STTAX				10, 30, 40	N	
70	Apply To	Unit Rate		COUNTYTA X				30, 40	N	

### Contents

[Hours-Based - For Calculation Purposes Only Rate Component](#)

Hours-Based - Dynamic Stepped SQ Rate Components  
 Hours-Based - Descriptions On Bill

### Hours-Based - For Calculation Purposes Only Rate Component

The strange thing about this rate is rate component 20. It is used “for calculation purposes only”. Notice what it’s calculating – the number of kilowatts (KW). Why is it necessary? Because hour-based steps have to be multiplied by the number of KW used by the customer in order to derive the real KWH based steps. Why? Because number of KWH / number of KW = number of hours. Therefore an hour \* number of KW = number of KWH.

### Hours-Based - Dynamic Stepped SQ Rate Components

Notice that rate components 30 and 40’s step boundaries are multiplied by the amount calculated by rate component 20 (i.e., the number of KW).

### Hours-Based - Descriptions On Bill

The following table illustrates how you might want to define the Description on Bill for these rate components.

Seq #	Description on Bill	Print on Bill
10	Monthly service charge of %R %D	Y
20	Calculate only rate to determine Hourly charge step multiplication factor.	N
30	First %Q hours of usage at %I per kWh %F	Y
40	Remaining %Q hours of usage at %I per kWh %F	Y
50	Subtotal	Y
60	State sales tax %R %T	Y
70	County use tax %R per kWh %T	Y

### Step Boundaries Based On The Number Of Days Of Usage And The Region In Which The Customer Resides

The following table illustrates the rate components necessary to levy a simple rate that contains the following charges:

- Monthly service charge: \$10.00.
- Charge \$.0832 for the first “x” kWh. X is equal to the number of days in the billing period multiplied by a factor that varies depending on where the customer lives.
- Charge \$.1112 for the remaining kWh.
- State Tax at the prevailing rate (this tax is a percentage of a dollar value).
- County Use Tax at the prevailing rate (this tax is an amount per kilowatt hour).

Seq #	RC Type	Value Type	Value	Bill Factor	Item Type	UOM/ TOU/ SQI	Step Range	Cross Reference	Calc Only	Result Type
10	Flat Charge	Charge	10.00						N	

20	Service Quantity	Unit Rate		BASEQTY		DAYS			Y	Unit Rate
30	Service Quantity	Unit Rate	0.0832			KWH	0 to 1 Multiply by RC: 20		N	
40	Service Quantity	Unit Rate	0.1112			KWH	1 to 99999999 Multiply by RC: 20		N	
50	Summary							30, 40	N	
60	Apply To	Percent		STTAX				10, 30, 40	N	
70	Apply To	Unit Rate		COUNTYTA X				30, 40	N	

## Contents

- [Dynamic Steps - SQ Rule On The Rate Schedule](#)
- [Dynamic Steps - For Calculation Purposes Only Rate Component](#)
- [Dynamic Steps - Bill Factors](#)
- [Dynamic Steps - Dynamic Stepped SQ Rate Components](#)
- [Dynamic Steps - Descriptions On Bill](#)

## Dynamic Steps - SQ Rule On The Rate Schedule

This rate's rate schedule must have an SQ rule that calculates the number of days in the rate's bill period. This SQ rule will add a new service quantity to the measured service quantities supplied to the rate by billing. This new service quantity would be labeled with an SQI of **DAY**S and would contain the number of days in the bill period. This service quantity is needed by rate component 20.

## Dynamic Steps - For Calculation Purposes Only Rate Component

The strange thing about this rate is rate component 20. It is used "for calculation purposes only". Notice what it's calculating – the number of days (SQI = **DAY**S) multiplied by some factor that differs based on the region (Bill factor = **BASEQTY**).

## Dynamic Steps - Bill Factors

Because the daily amount differs based on the region in which the customer resides, you'll need a bill factor – **BASEQTY** – with a characteristic that matches the region. This characteristic will be specified on the customer's premise. Then, at billing time, the system extracts the appropriate base quantity based on where the customer lives and multiplies it times the number of days in the customer's bill. The step boundaries are then multiplied by the resultant value (notice that they are set to 1).

## Dynamic Steps - Dynamic Stepped SQ Rate Components

Notice that rate components 30 and 40's step boundaries are multiplied by the amount calculated by rate component 20 (i.e., amount based on the number of days of consumption).

## Dynamic Steps - Descriptions On Bill

The following table illustrates how you might want to define the Description on Bill for these rate components.

Seq #	Description on Bill	Print on Bill
10	Monthly service charge of %R %D	Y
20	Calculate only rate component to determine Step multiplication factor	N
30	First %Q kWh at %I per kWh %F	Y
40	Remaining %Q kWh at %I per kWh %F	Y
50	Subtotal	Y
60	State sales tax %R %T	Y
70	County use tax %R per kWh %T	Y

## Calculated Minimum Charges

The following table illustrates the rate components necessary to levy a simple rate that contains the following charges:

- First 300 kilowatt hours at \$0.0382 per kWh.
- Remaining kilowatt hours at \$0.0673 per kWh.
- Minimum charge of \$0.15 per day.
- State Tax at the prevailing rate (this tax is a percentage of a dollar value).
- County Use Tax at the prevailing rate (this tax is an amount per kilowatt hour).

Seq #	RC Type	Value Type	Value	Bill Factor	Item Type	UOM/TOU/SQI	Step Range	Cross Reference	Calc Only	Result Type
30	Service Quantity	Unit Rate	0.0382			KWH	0 to 300		N	
40	Service Quantity	Unit Rate	0.0673			KWH	300 to infinity		N	
50	Summary							30, 40	N	
52	Service Quantity	Unit Rate	.15			DAYS			Y	Charge
54	Minimum Charge			52				30, 40	N	
60	Apply To	Percent		STTAX				30, 40, 54	N	
70	Apply To	Unit Rate		COUNTYTA X				30, 40	N	

### Contents

[Calculated Minimums - SQ Rule On The Rate Schedule](#)

[Calculated Minimums - For Calculation Purposes Only Rate Component](#)

[Calculated Minimums - Description On Bill](#)

## Calculated Minimums - SQ Rule On The Rate Schedule

This rate's rate schedule must have an SQ rule that calculates the number of days in the rate's bill period. This SQ rule will add a new service quantity to the measured service quantities supplied to the rate by billing. This new service quantity would be labeled with an SQL of **DAYS** and would contain the number of days in the bill period. This service quantity is needed by rate component 52.

## Calculated Minimums - For Calculation Purposes Only Rate Component

The strange thing about this rate is rate component 52. It is used "for calculation purposes only". Notice what it's calculating – the number of days (SQI = **DAYS**) multiplied by 1 (i.e., the number of days in the billing period).

## Calculated Minimums - Description On Bill

The following table illustrates how you might want to define the Description on Bill for these rate components.

Seq #	Description on Bill	Print on Bill
30	First %Q kWh at %I per kWh %F	Y
40	Remaining %Q kWh at %I per kWh %F	Y
50	Subtotal	Y
52	Calculate only rate component to determine number of days	N
54	Minimum charge %S	Y
60	State sales tax %R %T	Y
70	County use tax %R per kWh %T	Y

## Stepped Taxes

**Warning!** This is a difficult example. You should take the time to understand it even if you don't have stepped taxes because it illustrates the power of the various rate algorithms.

The following table illustrates the rate components necessary to levy a simple rate that contains the following charges:

- \$0.10 per kWh.
- Service charge of \$6.00
- Taxes are stepped:
- 6.383% for anything up to \$10
- 9.89% for everything above \$10

Note that the dollar amount should be prorated if the bill is prorated.

Seq #	Description	RC Type	Value Type	Value	UOM/TOU/SQI	Cross Reference	Calc Only	Result Type

10	Energy Charge	Service Quantity	Unit Rate	0.10	KWH		N	
20	Customer Charge	Flat	Charge	6.00			N	
30	Total Taxable Revenue	Summary				10, 20	Y	Charge
40	Taxable Rev for 1st tax tier	Maximum	Charge	10.00		30	Y	Charge
50	Taxable Rev for 1 <sup>st</sup> tax tier	Summary				30, 40	Y	Charge
60	Taxable Rev for 2 <sup>nd</sup> tax tier	Apply To	Percent	-100.00		50	Y	Charge
70	Taxable Rev for 2 <sup>nd</sup> tax tier	Summary				30, 60	Y	Charge
80	Tax for 1 <sup>st</sup> tier	Apply to	Percent	6.383		50	N	
90	Tax for 2 <sup>nd</sup> tier	Apply to	Percent	9.89		70	N	

Notice that sequences 30, 40, 50, 60 are calculation-only rate components that are used to calculate the amount of revenue applicable for each tax rate. We'll let you work the math.

## How To Add A New Service Quantity Rule Algorithm

**Technical Information Below!** This section is meant for technicians only!

If the SQ rule algorithms that are supplied with the base package are not sufficient for your rating requirements, you will have to write a new SQ rule algorithm. This section describes how to do this. After you have followed the steps outlined below, you can then set up the control tables necessary to invoke this algorithm as described under [Setting Up Service Quantity Rules](#).

**Warning!** Do NOT attempt to set up a new service quantity algorithm until you have a very good understanding of what your meters measure and how your rates charge for this consumption. In addition, you should completely understand the concepts described in [UOM versus TOU versus SQI](#) before creating a new algorithm. In addition, you should completely understand the concepts described in [When And How To Manipulate Measured Consumption](#) before creating a new algorithm.

Follow the following steps to define a new service quantity (SQ) rule:

- Assign a two-digit identifier to the SQ rule. This identifier should begin with an **X** or **Y**. For example, you could give your first SQ rule the identifier **X1**, the second **X2**, etc.
- Create an algorithm type for your new SQ rule. This algorithm type exists to define the types of parameters supplied to your SQ rule. Refer to [Base Package SQ Rules](#) for examples. This algorithm type must be identified by the two-digit identifier allocated above.

While SQ rules require an algorithm type, they do NOT require an algorithm. Why? Because you set up an SQ rule to define the values of the parameters passed to your program (and therefore you don't need an algorithm as algorithms exist to define the values of the parameters passed to a program).

- Go to [Look Up Options](#) and add a new lookup value for the field **SQ\_RULE\_FLG**. This should have a Field Value equal to the two-digit identifier allocated above.

- Write your new program. This program should be named **CMPRQ'nn'X** – where ‘nn’ is the same two-digit identifier. When you create the new program, please copy an existing SQ rule program (their program names are all in the format CIPRQ\_\_X). This way you will automatically inherit the copybooks necessary to support the routine.

**Warning!** Do NOT alter any of the copybooks contained in the linkage section of the COBOL algorithm routine. If you do need extra information for your algorithm you should execute SQL to get the information. If you do alter a linkage section copybook the upgrade process will be greatly complicated. Note, however, that virtually all information available to the rate application is being passed to the rule algorithms – so it is likely the information you need will be at hand.

## How To Add A New Register Rule Algorithm

**Technical Information Below!** This section is meant for technicians only!

If the register rule algorithms that are supplied with the base package are not sufficient for your rating requirements, you will have to write a new register rule algorithm. This section describes how to do this. After you have followed the steps outlined below, you can then set up the control tables necessary to invoke this algorithm as described under [Setting Up Register Rules](#).

**Warning!** Do NOT attempt to set up a new Register algorithm until you have a very good understanding of what your meters measure and how your rates charge for this consumption. In addition, you should completely understand the concepts described in [UOM versus TOU versus SQL](#) before creating a new algorithm. In addition, you should completely understand the concepts described in [When And How To Manipulate Measured Consumption](#) before creating a new algorithm.

Follow the following steps to define a new Register (register) rule:

- Assign a two-digit identifier to the register rule. This identifier should begin with an **X** or **Y**. For example, you could give your first register rule the identifier **X1**, the second **X2**, etc.
- Create an algorithm type for your new register rule. This algorithm type exists to define the types of parameters supplied to your register rule. Refer to [Base Package Register Rules](#) for examples. This algorithm type must be identified by the two-digit identifier allocated above.

While register rules require an algorithm type, they do NOT require an algorithm. Why? Because you set up a register rule to define the values of the parameters passed to your program (and therefore you don't need an algorithm as algorithms exist to define the values of the parameters passed to a program).

- Go to [Look Up Options](#) and add a new lookup value for the field **REG\_RULE\_ALG\_FLG**. This should have a Field Value equal to the two-digit identifier allocated above.
- Write your new program. This program should be named **CMPRR'nn'X** – where ‘nn’ is the same two-digit identifier. When you create the new program, please copy an existing register rule program (their program names are all in the format CIPRR\_\_X). This way you will automatically inherit the copybooks necessary to support the routine.

**Warning!** Do NOT alter any of the copybooks contained in the linkage section of the COBOL algorithm routine. If you do need extra information for your algorithm you should execute SQL to get the information. If you do alter a linkage section copybook, the upgrade process will be greatly complicated. Note, however, that virtually all information available to the rate application is being passed to the rule algorithms – so it is likely the information you need will be at hand.

# Quotes

A quote allows you to show a customer how their bill segments will look if a given set of pricing terms is applied to a given amount of consumption. If the customer accepts a quote, one or more real service agreements are created.

Refer to [Defining Quotation Options](#) for how to setup the system to enable quotation functionality.

**Separate module.** Please note that the quoting 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 topics in this section describe the quotation functionality.

## Contents

- [The Big Picture Of Quotations](#)
- [Maintaining Quotes](#)

## [The Big Picture Of Quotations](#)

The topics in this section provide background information about a variety of quotation features and functions.

**Warning!** If you are not familiar with the concepts described under [Sales and Marketing](#), [Billing, Rates](#) and [Interval Billing](#) (assuming you have quotes generated using interval consumption) the quotation functionality will be difficult to understand.

## Contents

- [Basic Quotation Concepts](#)
- [Printing Quotes](#)

## [Basic Quotation Concepts](#)

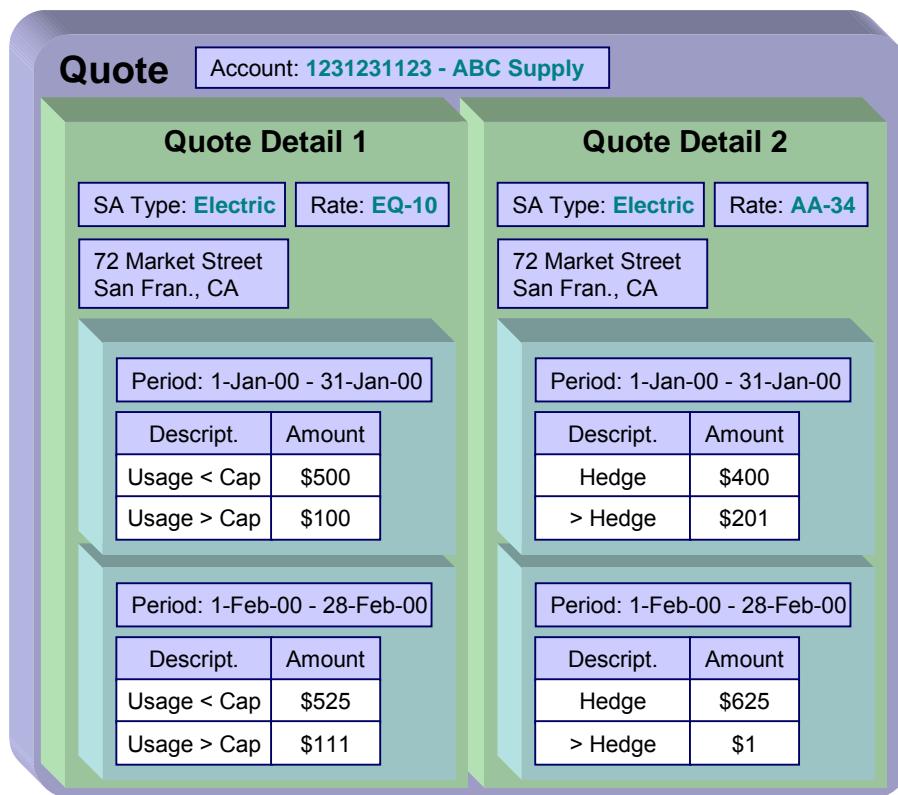
The section describes basic quotation concepts.

## Contents

- [A Quote Shows How Bill Segments Might Look](#)
- [Proposal Service Agreements Must Exist](#)
- [Creating Quotes And Quote Details](#)
- [Completing A Quote Creates Quote Routings \(And More\)](#)
- [Accepting / Declining Quote Details](#)
- [Tinkering With Proposal SAs After Quote Generation](#)
- [Closing Quotes](#)
- [Alerts](#)
- [Trees](#)

## A Quote Shows How Bill Segments Might Look

A quote shows how a customer's bill segments will look if a given set of pricing terms is applied to a given amount of consumption. The following example illustrates how a quote might look that provides a customer with two different options for their electric service. If the customer elects either option, a new service agreement would be created.



Please note the following about a quote:

- A quote is related to an account.
- A quote can contain many quote details. A quote detail shows how bill segments will look if a given set of pricing terms (e.g., rates, riders, contract quantities, time-of-use-maps, contract options, etc.) is applied to a given amount of consumption.
- A quote detail may contain many simulated bill segments. For example, a quote detail could show the customer what their monthly bills would be if a given set of pricing terms were used over the course of several months (where each month has different consumption levels and different prices).
- A quote may contain multiple quote details for the same service. For example, a quote can contain four different pricing options for the same service (where each option is shown using a separate quote detail). The customer can then compare them and choose the optimum option.
- A quote may contain quote details for multiple premises and multiple services at each premise. For example, a quote can contain quote details for electric and gas service at multiple premises.

## Proposal Service Agreements Must Exist

You cannot create a quote unless its account has at least one [proposal](#) service agreement (SA). A proposal SA contains the pricing terms, legal terms, and template consumption used to generate the simulated bill segments that appear on a quote. It might be helpful to view a proposal SA as special type of service agreement that is ignored when a bill is produced for an account.

The topics in this section provide detailed information about proposal SAs.

### Contents

- [Proposal SAs Contain Pricing Terms](#)
- [Legal Terms and Conditions May Be Specified On Both Proposal And Real SAs](#)
- [Proposal SAs Contain Billing Scenarios And Template Consumption](#)
- [Proposal SAs Differ From Real SAs](#)
- [Proposal SA State Transition](#)
- [Orders Are Used To Create Proposal SAs](#)
- [Proposal SAs May Be Grouped Under An Umbrella Agreement](#)
- [Proposal SAs And Interval Consumption](#)

#### Proposal SAs Contain Pricing Terms

The pricing terms (e.g., rates, contract quantities, riders, bill factor values, load curves, TOU maps, etc.) used to generate a quote detail's simulated bill segments reside on a special type of service agreement (SA) referred to as a "proposal SA". Proposal SAs are like real SAs in many respects. In fact, you can view and change a proposal SAs pricing terms using the same transaction used to maintain real service agreements (i.e., [service agreement maintenance](#)).

**Quote details are rather empty.** A quote detail doesn't contain the pricing terms or template consumption. Rather, a quote detail references a proposal SA and the proposal SA contains the pricing terms and template consumption.

#### Legal Terms and Conditions May Be Specified On Both Proposal And Real SAs

In addition to containing pricing terms, proposal SAs also contain the legal terms and conditions (T&Cs) that govern the proposal.

<b>Proposal Service Agreement</b>							
SA Type: <b>Electric</b>							
72 Market Street San Fran., CA							
Rate: <b>EQ-10</b>							
<table border="1"> <thead> <tr> <th>Quantity Type</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>Subscribed Demand</td> <td>10</td> </tr> <tr> <td>Hedge Percentage</td> <td>15</td> </tr> </tbody> </table>		Quantity Type	Value	Subscribed Demand	10	Hedge Percentage	15
Quantity Type	Value						
Subscribed Demand	10						
Hedge Percentage	15						
<b>Terms &amp; Conditions</b>							
<p>This quote requires a commitment of at least 6 months. Early cancellation results in a fee of 2 times the average bill amount.</p>							
<p>This quote was calculated using historical market prices. The historical market prices should not be assumed to be the same in the future</p>							

Your master list of terms and conditions are defined when you setup your control tables. The system defaults a service agreement's T&Cs from the [start option](#) used to initially create it. You can change a SA's T&Cs at will.

**Real SAs and proposal SAs can reference T&Cs.** It should be noted that both proposal and real SAs can reference T&Cs. The T&Cs on a proposal SA are those that will apply if the customer accepts the related quote detail. The T&Cs on a real SA are those that apply while the SA is active.

### Proposal SAs Contain Billing Scenarios And Template Consumption

In addition to the pricing terms and the legal terms and conditions, a proposal SA also contains [billing scenarios](#). A billing scenario defines:

- The number of simulated bill segments that will be generated when a quote detail is produced for the proposal SA.
- The bill period of each simulated bill segment.
- The amount of consumption to be priced under each simulated bill segment. Note this statement is only true for non-interval consumption. Refer to [Proposal SAs And Interval Consumption](#) for a description of how interval consumption is defined on a proposal SA.

A proposal SA's billing scenarios can be created as follows:

- A plug-in spot exists on [SA type](#) called **Proposal SA Creation**. Algorithms plugged into this spot can setup billing scenarios. Refer to [Enabling The Automatic Generation Of Billing Scenarios](#) for more information about how to do this.

- A user can add / change / delete billing scenarios on a proposal SA using [Service Agreement - Billing Scenario](#).
- You could design a workflow process to initiate a request to the customer to send their template consumption to you. When the customer sends this information to you (via a notification), the proposal SA's billing scenarios can be setup. Refer to [Workflow and Notifications](#) for more information.

### Proposal SAs Differ From Real SAs

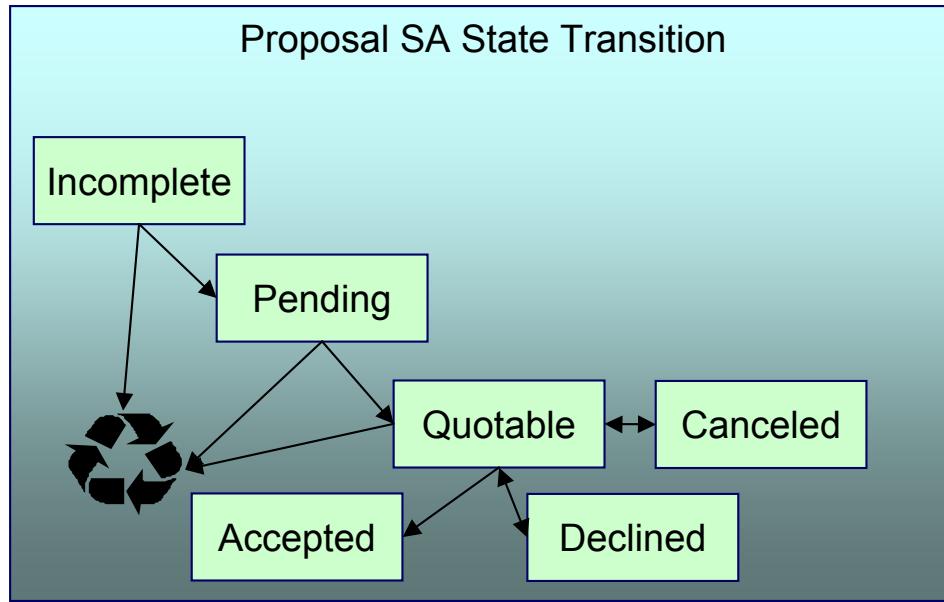
A proposal SA is just like any other SA in that it contains the pricing terms that are used to generate how much the customer owes for service. In fact, the [service agreement maintenance transaction](#) is used to maintain both types of service agreements (however, you cannot use this transaction to create a proposal SA; proposal SAs can only be created using the [order transaction](#)).

However, there are differences between proposal SAs and real SAs:

- Billing ignores proposal SAs.
- Proposal SAs cannot have financial transactions.
- Proposal SAs are not subject to credit and collections.
- Proposal SAs do not appear on lists of “active” SAs in the system’s various zones, grids and trees.
- Fieldwork is not performed for proposal SAs.
- A proposal SA’s state transition is very different from a real SA. Refer to [The Lifecycle Of A Service Agreement](#) for an illustration of a real SAs state transition. Refer to [Proposal SA State Transition](#) for an illustration of a proposal SAs state transition.

### Proposal SA State Transition

The following diagram shows the possible lifecycle of a proposal SA.



You should never see an **Incomplete** proposal SA as this is a transitory state used when a proposal is first initiated using the [Order](#) transaction. The system always transitions such proposal SAs to the **Pending** state. Please note that when a proposal SA becomes **Pending**, the system executes any **Proposal SA Creation** algorithms plugged in on the proposal SA's SA type. You might use such algorithms to generate billing scenarios or initiate a workflow process that requests template consumption from the prospect.

A proposal SA exists in the **Pending** state while you are working on the proposal. For example, you might want to leave the proposal SA in this state while you wait for the prospect to send you their historical consumption.

A proposal SA is **Quotable** when a quote detail can be generated for it (at least one billing scenario must be linked to the proposal SA before it can become **Quotable**). Note, proposal SAs can be modified when they are **Quotable**.

**Automatically transitioning a proposal SA to the Quotable state when it's first created.** As described earlier, you can setup a **proposal SA creation** algorithm to automatically create billing scenarios when a proposal SA is created. The base package algorithm can also automatically transition the proposal SA to the **Quotable** state (thus making it ready for quotation). Refer to [Enabling The Automatic Generation Of Billing Scenarios](#) for more information.

A proposal SA becomes **Accepted** when a customer accepts one of its quote details.

A proposal SA becomes **Declined** when a customer declines one of its quote details (or its quote expires). A **Declined** proposal SA can be “reopened” (by making it **Quotable**). You may need to do this if you want to change a proposal SA’s terms in preparation of generating a new quote detail.

A proposal SA becomes **Canceled** when a quote detail is withdrawn (or its quote expires). You may need to do this if you want to change a proposal SA’s terms in preparation of generating a new quote detail.

**Incomplete**, **Pending** and **Quotable** proposal SAs may be deleted.

## Orders Are Used To Create Proposal SAs

Proposal SAs can only be created using the [order](#) transaction. This transaction allows a user to pose eligibility-oriented questions to a customer or prospect. The system determines which [packages](#) of goods and services for which they are eligible based on the answers to the questions. If the customer is interested in receiving a package, one or more SAs (real and/or proposal) are created using the [SAs To Create](#) information defined on the selected package.

**You must setup campaigns and packages before you can create an order.** If you are not familiar with the [order](#) transaction, you should take the time to form an intuitive understanding of the concepts described under [Sales & Marketing](#). With this knowledge in place, you can design and set up the campaigns and packages necessary to allow an order to create proposal SAs. Keep in mind that a single switch controls whether real SAs or a proposal SAs are created when a package is selected on the order transaction. This switch resides on [Package - SAs To Create](#).

**A package could create a combination of real and proposal SAs.** When a customer elects to take a package of goods and services (via the [order](#) transaction), the system creates a service agreement for each SA type / start options defined on the package's [SAs To Create](#). It is possible to setup a package to create both real and proposal SAs. For example, the customer may be calling you to start electric service, but they may also be interested in seeing a proposal for gas service.

**You don't create special SA types for proposal SAs.** The SA type on a proposal SA is the same as the SA type on the real SA that's created when the proposal SA is accepted. This is because the business rules on your SA types govern both real and proposal SAs.

**You don't need special start options for proposal SAs.** The start options that you setup to default information onto new, real service agreements are also referenced on the [SA To Create](#) information used to create proposal SAs.

## Proposal SAs May Be Grouped Under An Umbrella Agreement

Each proposal SA may be part of a larger contract for multiple services that you are negotiating with a customer or group of customers. You may use an [umbrella agreement](#) and its collection of terms of service records to group together multiple proposal service agreements.

## Proposal SAs And Interval Consumption

**Assumption.** This section assumes you are very comfortable with the concepts described under [Interval Billing](#).

Service agreements with interval data have rates that price the interval consumption using time-of-use prices and / or direct prices. Proposal SAs for interval service agreements require billing scenarios to define the time period of each simulated bill segment on the quote detail. However, these billing scenarios typically won't have template consumption (unless regular meters are used in addition to interval meters on the service agreement). Rather, interval billing proposal SAs require interval profiles and datasets, time-of-use maps, and contract options because their rates use this information to calculate the simulated bill segments.

The following points provide an overview of how interval information can be populated on proposal SAs:

- As described under [Orders Are Used To Create Proposal SAs](#), a proposal SA is populated using the start option defined on the package specified when an order is completed (note, a package can cause multiple proposal SAs to be created where each proposal SA references a different start option).
- Start options can reference all types of interval information (interval profiles, time-of-use maps and contract options). If you're familiar with interval billing concepts, you know that the system treats "common" interval information differently from "SA-owned" information when a start option is used to create a service agreement:
  - "Common" interval information referenced on a start option is simply copied to the newly created service agreement. This is exactly what you want. For example, when you create a proposal SA using a start option that references a "deemed profile" or a "standard time-of-use (TOU) map", you don't want a new deemed profile or TOU map created. Rather, you want the start option's deemed profile and time-of-use map to be referenced on the proposal SA.
  - "SA Owned" interval information on a start option doesn't work this way. Rather, the system creates new interval information and references these on the newly created service agreement. For example, if a start option indicates the service agreement should have an SA-owned interval profile for actual consumption and another for derived consumption (e.g., excess demand), the system creates two interval profiles and links them to the new service agreement. For real SAs, this logic is perfect. For proposal SAs, this logic is perfect for interval profiles that are derived during billing. However, for interval profiles that contain actual consumption, this isn't what you want. This is because, for actual consumption on proposal SAs, you just need to reference an existing profile whose consumption can be used as the template consumption. The next point provides a few options.

**Note.** Although the above describes issues related to SA owned interval profiles, similar issues may apply for SA owned TOU maps or to contract options that are specific to a specific service agreement. To successfully generate a quote for a customer, you may need to create template data for these entities as well.

- You can choose from the following methods to populate a proposal SA with "actual consumption":
  - You could design a workflow process that asks the customer to send their template consumption to you. This could be done by creating a [Proposal SA Creation](#) plug-in on the proposal SA's [SA type](#) (note, as of the current release, this algorithm does not exist in the base package, but it would be very simple to write). When the customer sends this information to you (via a notification), the datasets on the interval profile that contain the proposal SA's simulated actual consumption can be setup. Refer to [Workflow and Notifications](#) for more information.

- You could ask the user (during the order taking process) to define the type of consumption profile (e.g., are they a doctor's office, bakery, laundry, etc). If you're familiar with campaigns and orders, you'll know that you can do this by setting up a question on the campaign. For example, you could design the campaign to pose a question like, "Please choose the consumption profile to be used when the quote is generated". Then, based on the user's answer, you could set up the proposal SA using profiles consistent with the selected type. The following would be required to setup this type of question:
  - Create a "substitute" start option for each consumption profile. Each start option will contain the normal common and derived interval profiles that are used for a real interval SA. The interval profile on the start option that's used for actual consumption would also use a common profile. This common profile would contain the "template" actual consumption that will be used in place of the customer's actual consumption. This template consumption can be used on any interval profile created using this start option.
  - Setup a campaign that asks a user to identify the consumption profile to be used to generate the quote as described above.
  - Setup a package under the campaign for each consumption profile. Each package will have eligibility rules to indicate that it is only selectable if the respective consumption profile is chosen on the order.
  - Each package will also reference the "substitute" start option that corresponds with its consumption profile.
  - At this point, you have everything setup to create a proposal SA with a consumption profile that can be used to generate simulated bill segments (remember, the proposal SA's SA type needs a ***Proposal SA Creation*** algorithm that [creates billing scenarios](#)). If the customer accepts this proposal SA, you need to setup a real SA with the correct profiles (you don't want the consumption profile on the real SA, nor do you want any derived profiles that were created when the simulated bill segments were generated). The remaining points describe how to do this.
  - You want these proposal SAs to contain a characteristic that defines the true start option that should be used to create the real SA. To do this, simply define a default char type / value on the "substitute" start option that contains the real start option (this default information is copied onto the proposal SA when it's created). **Note.** The SA type of the "real" start option should be the same as the proposal SA type.
  - As described under [Accepting / Declining Quote Details](#), you must plug-in a ***Proposal SA Accepted*** algorithm that has a parameter of the char type on the proposal SA that contains the start option to use when the real SA is created. It then does the work for you. Refer to [PSAA-PS](#) for more information about this algorithm.
- Other solutions are only limited by your imagination.

## Creating Quotes And Quote Details

Once an account has ***quotable*** proposal SAs, you can generate a quote for the account using the quote maintenance transaction. When you generate a quote for an account, you can include quote details for any / all of the account's ***quotable*** proposal SAs.

The logic used to create a quote detail's simulated bill segments is not hard-coded. Rather, the system calls the algorithm plugged in on the proposal SA's ***SA type*** in the ***Proposal SA - Bill Segment Generation*** plug-in spot. Refer to [Enabling The Generation Of Simulated Bill Segments](#) for information about this plug-in spot.

The major difference between a bill segment linked to a quote detail and one linked to a bill is that a quote detail's bill segments are calculated using simulated consumption. This means that the financial information on quote details is not a guess – it's what the bill would be if a given amount of consumption was priced using the pricing terms (as long as you opt to use rate application to calculate the simulated bill segments).

In addition to this major difference, there are a few minor differences between a real bill segment and a simulated bill segment:

- A quote detail bill segment does not have a foreign key to a bill. Rather, it references a quote detail.
- A quote detail bill segment's prime key is inherited from its quote detail (rather than its bill).
- A quote detail bill segment can never have a financial transaction linked to it.
- A quote detail bill segment doesn't have read details (unless you perform TOU mapping of interval data).
- A quote detail bill segment's state transition is simpler (it's either in the **Error** state or **OK**).

## Completing A Quote Creates Quote Routings (And More)

When a quote's quote details are all **OK**, you change the status of the quote to **complete** to indicate the quote is ready to be sent to the customer or prospect.

When a quote is completed, the system creates a [quote routing](#) for each person who receives a copy of the account's quotes. When you indicate a person should receive a copy of the account's quotes, you define the routing method (e.g., email, fax, post) and the appropriate address.

**Recipients of quotes.** You define who receives a copy of an account's quotes on the [Account - Persons](#) page. The system assumes (by default) that the main person linked to the account should receive the account's quotes. You can subsequently indicate any person linked to the account should receive its quotes (and turn off the creation of quotes for the main person). In addition, you can manually add additional quote routings to a quote if you need to send the quote to individuals not currently linked to the account. Refer to [How To Reprint A Quote](#) for more information.

The information that appears on the quote routing record is controlled by the [quote route type](#) specified on the respective account / person information:

- If the quote route type indicates the person's quotes are routed via fax, the person's fax number is used as the routing information. The system knows which of a [person's phone numbers](#) is a fax number by the phone type.
- If the quote route type indicates the person's quotes are routed via email, the person's [Email address](#) is used as the routing information.
- If the quote route type indicates the person's quotes are routed via the postal service, the address constituents are populated with the address specified on the bill address source that is also specified on the [account / person information](#). Note, if the person has a [seasonal address](#) effective on the business date, the seasonal address will be used regardless of the value of bill address source.

Regardless of whether the quote is routed via fax, email or post; the routing record is also stamped with the batch process and run number that performs the download of the quote information. The batch process is defined on the quote route type. Refer to [Printing Quotes](#) for more information about this batch process.

In addition to creating routing information at completion time, the system executes any quote completion algorithms plugged in on the account's customer class. For example, you may want to create a workflow process to manage the processing of the quote. To do this, you plug in a Quote Completion Algorithm that creates a workflow process on the respective [customer class](#).

## Accepting / Declining Quote Details

A customer can accept or decline each quote detail on a quote.

When a quote detail is accepted, the system:

- Changes the state of the related proposal SA to **accepted**.
- Executes the **Proposal SA - Acceptance** algorithms plugged in on the proposal SA's SA type. The base package algorithm creates a real SA by copying the information from the proposal SA. The real SA is created in the **pending start** state. This new SA behaves similarly to a **pending start** SA that's created via the [Order](#) or [Start/Stop](#) transactions (e.g., field activities are initiated if the SA type has been setup accordingly).

When a quote detail is declined, the user must supply a decline reason. The system then changes the state of the related proposal SA to **declined**.

Refer to [Proposal SA State Transition](#) for more information about a proposal SA's states.

**Reopening a declined proposal SA.** If a customer changes their mind about a **declined** proposal SA, you can return it to the **quotable** state using the [service agreement maintenance transaction](#).

**Rejecting an entire quote.** If all details on a quote are being declined, you can simply change the state of the quote to **closed** rather than reject each quote detail. Refer to [Closing Quotes](#) for more information.

## Tinkering With Proposal SAs After Quote Generation

The system allows a **quotable** proposal SA to be changed, EVEN IF QUOTE DETAILS HAVE BEEN GENERATED. This is intentional as otherwise you would have to create an entirely new proposal SA if you wanted to send another quote for slightly different terms. Your technical team could always introduce a user exit to prevent this from happening if you really didn't want this to take place.

Because the system allows this, a potential issue is that the customer may end up accepting a proposal SA that differs from the terms on his quote (e.g., you could change the SA after the quote detail is generated). For this reason, we snapshot the version number of the proposal SA on the quote detail at the time it's generated. If a proposal SA is accepted and its version number differs from the value on its quote details, a warning is produced.

A warning is also issued if you change a proposal SA and a quote detail exists (reminding you to generate a new quote / quote detail).

In addition to the above, it's also important to understand that the system intentionally allows multiple quotes to reference a given proposal SA. This allows you to leave multiple quotes open that reference the same proposal SA(s).

## Closing Quotes

Every quote has an expiration date. The [CLOSEQTE](#) background process **closes** quotes on their expiration date. This process changes the state of all related proposal SAs to **canceled** or **declined** (an input parameter of this process controls which state is used).

You can also close a quote real time using the quote maintenance transaction. When you do this, the system asks if the related proposal SAs should be left as **quotable** or if they should be changed to **canceled** or **declined**.

**Reopening a declined proposal SA.** If a customer changes their mind about a **declined** proposal SA, you can return it to the **quotable** state using the [service agreement maintenance transaction](#).

## Alerts

The [alert zone](#) highlights if:

- The selected account has **pending** or **complete** quotes. Refer to [Quote Lifecycle](#) for more information.
- The selected account has **pending** or **quotable** proposal SAs. Refer to [Proposal SA State Transition](#) for more information.

## Trees

The [account tree](#) highlights:

- The existence of non-closed quotes
- The existence of proposal SAs and their related quote details

## Printing Quotes

The contents of this section describe the technical implementation of printing quotes.

### Contents

- [Quote Routings Are Created For Each Recipient Of A Quote](#)
- [Quote Route Types Control The Information Merged Onto Quotes](#)
- [Technical Implementation Of Online Quote Image](#)
- [Technical Implementation Of Printing Quotes In Batch](#)
- [Reproducing The Quote Print Flat File](#)
- [How To Reprint A Quote \(For The Original Recipients or For Someone New\)](#)
- [The Doc 1 Records](#)

## Quote Routings Are Created For Each Recipient Of A Quote

Refer to [Completing A Quote Creates Quote Routings](#) for a description of how the system constructs the information used to route a quote to one or more recipients.

## Quote Route Types Control The Information Merged Onto Quotes

Every quote routing references a [quote route type](#). The route type controls the following quote functions:

- It contains an algorithm that is responsible for extracting the information merged onto your quotes. Specifically, algorithms of this type create the “flat file records” that are passed to your quote-print software. Algorithms of this type are called under the following scenarios:
  - The background process that builds the flat file that’s passed to your quote-print software calls these algorithms to construct the “flat file records” for each quote. Refer to [Technical Implementation of Printing Quotes In Batch](#) for the details.
  - If your quote-print software has the ability to construct a real-time image of a quote (in a PDF), you can plug in an **Online Quote Image** algorithm on the [Installation Record](#). This type of algorithm calls the quote’s dispatch group extract algorithm to extract the information that is merged onto the quote. Refer to [Technical Implementation Of Online Quote Image](#) for the details.

Refer to [The Doc 1 Records](#) for examples of the record layouts passed to the Doc 1 print software.

- It contains the ID of the background process that builds the flat file that’s passed to your quote-print software. The base package example of this process (known by the batch control ID of **QUOTROUT**) simply calls the extract algorithm on the routing record’s route type to format the information placed onto the flat file.

## Technical Implementation Of Online Quote Image

Users can view an image of any quote if you set up the following:

- Plug in an **Online Quote Image** construction algorithm on the [Installation Record](#). Refer to [ONQD-DOC1](#) for an example of such an algorithm. Note if your quote-print software is not capable of producing a PDF containing an image of a quote, users will not be able to view images of quotes.
- Plug in the appropriate extract algorithm on each [quote route type](#). Algorithms of this type format the records that contain the information that is passed to your printing software. Refer to [QTEX-EX](#) for an example of such an algorithm.

When you plug in these algorithms, a button appears on [Quote - Main](#). When a user clicks this button, the following takes place:

- The installation record’s **Online Quote Image** construction algorithm is executed.
- This algorithm calls the quote’s route type’s extract algorithm. This algorithm constructs the information that’s merged onto the quote and returns it to the **Online Quote Image** algorithm. This algorithm, in turn, passes it to the quote-print software.
- The quote-print software renders an image of the quote in a PDF and returns it to the **Online Quote Image** algorithm.
- And finally, the **Online Quote Image** algorithm displays the PDF in a browser session. Note that the client must have Adobe Acrobat Reader installed to view PDF files.

## Technical Implementation Of Printing Quotes In Batch

The batch process that extracts quote information reads all quote routing records in a given run number that are marked with its batch control ID. For each quote routing record, it creates numerous records on a flat file. These records contain the information that is merged onto your quotes.

The base package example of this process (known by the batch control ID of [QUOTROUT](#)) simply calls the extract algorithm on the routing record's route type to format the information placed onto the flat file. Refer to [Quote Route Types Control The Information On Quotes](#) for more information.

**If your software doesn't support online quote images.** The algorithm that formats quote extract records that's plugged in on [quote route type](#) serves two purposes: 1) it formats the records used to construct online images of a quote, and 2) it formats the records downloaded to your quote print software in batch. If your quote printing software does not support the rendering of quote images real time, there is no need to create an extract algorithm. Rather, you should simply develop your own download process that both formats the extract records and downloads them (and then specify this batch process on your quote route type).

## Reproducing The Quote Print Flat File

You can reproduce the flat file at any time. Simply request the quote production background process and specify the run number associated with the historic run.

## How To Reprint A Quote (For The Original Recipients or For Someone New)

A copy of a quote is sent to every individual / business specified in the [quote's routing details](#). The system creates the routing details when a quote is completed. Refer to [Completing A Quote Creates Quote Routings](#) for more information about how this information is compiled during quote completion.

If you want to reprint a quote OR if you want to send a quote to someone other than the original recipients:

- Navigate to the [Quote - Main](#) page.
- Transfer to the [Quote - Quote Routings](#) tab.
- Insert a new row (click the + button). Note that you can only insert information on a quote that is **Complete**.
- Specify the name and address to which the quote should be sent. If you want to send the quote to someone other than the original recipient, specify the individual's name and address.
- Save the quote.

If a quote is **Complete** but the print process hasn't yet executed, you may change the name and address on the original routing details; you don't have to insert a new routing row.

## The Doc 1 Records

Numerous different types of records are interfaced to the Doc 1 quote print software. However, they all share a common structure:

- The first 4 bytes are called the **Printer Key**. This field is a record type used by the Doc 1 quote template.
- The next X bytes are called the **Sort Key**. This field is used to ensure the various records used to construct a printed quote appear in the correct order in the interface file.
- The next 12 bytes are called the **Mailing Key**. This field is used to control the order in which the formatted quotes are printed.
- The remaining bytes contain the **Quote Information** that appears on the printed quote. The type of information differs for each type of record.

The topics in this section describe each component.

## Contents

- [Printer Key \(Record Types\)](#)
- [Sort Key](#)
- [Mailing Key](#)
- [Quote Information](#)

### Printer Key (Record Types)

The following table defines each printer key (i.e., record type) on the quote print flat file.

Record Type	Description	Notes
0010	Global extract information	Required. 1 per flat file
0100	Quote record	Required. 1 per Quote.
0200	Quote Detail record	1 per quote detail.
0300	Terms and Conditions record	1 per terms and conditions linked to the proposal SA.
0400	Bill segment record	1 per bill segment linked to the quote detail.
0500	Bill calculation header	1 per bill calculation header linked to the bill segment.
0600	Bill calculation line	1 per bill calculation line linked to the bill calculation header.
0700	End of record type 0500	1 for each bill calculation header
0800	Read detail	1 for each meter read detail linked to the bill segment.
0900	SQ detail	1 for each service quantity detail linked to the bill segment.
1000	End of record type 0400	1 for each bill segment.
1100	End of record type 0200	1 for each quote detail.
9999	End of record type 0100	1 for each quote.

### Sort Key

The following table defines the structure of the sort key that appears on each quote print record. Please note that different components are filled in for each printer key.

Field Name	Format	Source/Value/Description
Quote ID	A12	
Sequence Number	N2	The sequence number of the account / person information associated with this quote routing information.
Quote record group	A2	10 Quote 20 Quote Detail 30 Quote end
Quote Detail ID	A12	
Quote Detail record group	A2	10 Quote Detail record 20 SA Terms and Conditions 30 Bill Segment 40 Quote Detail record end
SA T&C Short Description	A60	
Bill Segment End Date	A10	
Bill Segment Id	A12	
Bill Segment Group	A2	10 Bill segment 20 Bill segment calculation group 30 Bill segment read detail 40 Bill segment SQ detail 50 Bill segment end
Bill Calc Header Sequence	N2	
Bill Calc Header Group	A2	10 Bill calculation header 20 Bill calculation line 30 Bill calculation end
Line Seq	N4	Defines order of rows for similar record types

### Mailing Key

The following table defines the structure of the mailing key that appears on each quote print record.

Field Name	Format	Source/Value/Description
Postal code	A12	This is the postal code on the quote routing record.

### Quote Information

The topics in this section describe the information that appears on each bill print record.

### Contents

[Global Extract Information \(0010\) Record Definition](#)

[Quote \(0100\) Record Definition](#)  
[Quote Detail \(0200\) Record Definition](#)  
[SA Terms and Conditions \(0300\) Record Definition](#)  
[Bill Segment \(0400\) Record Definition](#)  
[Bill Calculation Header \(0500\) Record Definition](#)  
[Bill Calculation Line \(0600\) Record Definition](#)  
[End of Bill Calculation Header \(0700\) Record Definition](#)  
[Bill Segment Read \(0800\) Record Definition](#)  
[Bill Segment SQ \(0900\) Record Definition](#)  
[End of Bill Segment \(1000\) Record Definition](#)  
[End of Quote Detail \(1100\) Record Definition](#)  
[End of Quote \(9999\) Record Definition](#)  
[Address Sub-record Definition](#)

[Global Extract Information \(0010\) Record Definition](#)

This is a “header” record that contains information about the extract run as a whole.

**Note.** Please be aware that if you do not sort your extract file by **CIS Sort Key** after it is produced, this record will physically be the last record on the extract file.

Field Name	Format	Source/Value/Description
BATCH_CD	A8	Batch Control Code of the extract process.
BATCH_NBR	N10	Batch Number of the extract process
BATCH_RERUN_NBR	N10	Batch Rerun number of the extract process.
QTE_CNT	N10	Count of quotes on this extract.
EXTRACT_DTTM	A26	System time of extraction.
QUOTE_ID_START	A12	This is only used when a quote is produced as a result of an online request (from Quote – Main). It contains the Quote Id to extract.
QUOTE_ID_END	A12	This is only used when a quote is produced as a result of an online request (from Quote – Main). It contains the Quote Id to extract.
THREAD_NBR	N10	This field is only used if quotes are extracted in batch. It contains the thread number in which the quote was extracted.
THREAD_CNT	N10	This field is only used if quotes are extracted in batch. It contains the total number of threads that were extracted.

[Quote \(0100\) Record Definition](#)

Field Name	Format	Source/Value/Description
QUOTE_ID	A12	CI_QUOTE
ACCT_ID	A10	CI_QUOTE
EXPIRE_DTTM	A26	CI_QUOTE
QUOTE_STAT_FLG	A2	CI_QUOTE
CURRENCY_CD	A3	CI_ACCT
CUR_SYMBOL	A4	CI_CURRENCY_CD
ENTITY_NAME1	A64	CI_QUOTE_RTG. Mailing Entity Name1.

<b>Field Name</b>	<b>Format</b>	<b>Source/Value/Description</b>
ENTITY_NAME2	A64	CI_QUOTE_RTG. Mailing Entity Name2.
ENTITY_NAME3	A64	CI_QUOTE_RTG. Mailing Entity Name3.
ADDRESS_SBR	A361	CI_QUOTE_RTG. Refer to its components in a later section.
EMAILID	A70	CI_QUOTE_RTG
PHONE-FAX	A64	CI_QUOTE_RTG

#### Quote Detail (0200) Record Definition

<b>Field Name</b>	<b>Format</b>	<b>Source/Value/Description</b>
QUOTE_DTL_ID	A12	CI_QUOTE_DTL
SA_ID	A10	CI_QUOTE_DTL
TOTAL_AMT	N13.2	CI_BSEG_CALC. Sum of all calculation amounts (quote detail's bill segments).
PREMISE_ID	A10	CI_SA (Characteristic Premise)
ADDRESS_SBR	A361	CI_PREM. Refer to its components in a later section.
SA_TYPE_DESCR	A60	CI_SA_TYPE_L
SVC_TYPE_CD	A2	CI_SVC_TYPE_L
SA_VERSION	N5	CI_QUOTE_DTL
QTE_DTL_SA_VERSION	N5	CI_QUOTE_DTL
RS_DESCR	A60	CI_RS_L. Description of Rate effective on SA start date.

#### SA Terms and Conditions (0300) Record Definition

<b>Field Name</b>	<b>Format</b>	<b>Source/Value/Description</b>
START_DT	A10	CI_SA_TC
END_DT	A10	CI_SA_TC
TC_TYPE_CD	A12	CI_TC
DESCR	A60	CI_TC_L
TC_TYPE_TXT	A2000	CI_TC_L

#### Bill Segment (0400) Record Definition

<b>Field Name</b>	<b>Format</b>	<b>Source/Value/Description</b>
START_DT	A10	CI_BSEG
END_DT	A10	CI_BSEG
BSEG_STAT_FLG	A2	CI_BSEG
BSEG_AMT	S13.2	CI_BSEG_CALC_LN

#### Bill Calculation Header (0500) Record Definition

<b>Field Name</b>	<b>Format</b>	<b>Source/Value/Description</b>
START_DT	A10	CI_BSEG_CALC

<b>Field Name</b>	<b>Format</b>	<b>Source/Value/Description</b>
END_DT	A10	CI_BSEG_CALC
RS_CD	A8	CI_BSEG_CALC
CALC_AMT	S13.2	CI_BSEG_CALC
DESCR_ON_BILL	A80	CI_BSEG_CALC

**Bill Calculation Line (0600) Record Definition**

<b>Field Name</b>	<b>Format</b>	<b>Source/Value/Description</b>
APP_IN_SUMM_SW	A1	CI_BSEG_CALC_LN
CALC_AMT	S13.2	CI_BSEG_CALC_LN
EXEMPT_AMT	S13.2	CI_BSEG_CALC_LN
UOM_CD	A4	CI_BSEG_CALC_LN
TOU_CD	A8	CI_BSEG_CALC_LN
SQI_CD	A8	CI_BSEG_CALC_LN
BILL_SQ	S12.6	CI_BSEG_CALC_LN
DESCR_ON_BILL	A80	CI_BSEG_CALC_LN

**End of Bill Calculation Header (0700) Record Definition**

<b>Field Name</b>	<b>Format</b>	<b>Source/Value/Description</b>
Dummy field		

**Bill Segment Read (0800) Record Definition**

<b>Field Name</b>	<b>Format</b>	<b>Source/Value/Description</b>
SP_ID	A10	CI_BSEG_READ
BADGE_NBR	A16	CI_MTR
UOM_CD	A4	CI_BSEG_READ
TOU_CD	A8	CI_BSEG_READ
SQI_CD	A8	CI_BSEG_READ
RDG_DEC_POS	N1	CI_REG. Decimal positions of the reads.
START_REG_READING	S9.6	CI_BSEG_READ
START_READ_DTTM	A26	CI_BSEG_READ
START_ESTIMATE_SW	A1	CI_REG_READ
END_REG_READING	S9.6	CI_BSEG_READ
END_READ_DTTM	A26	CI_BSEG_READ
END_ESTIMATE_SW	A1	CI_REG_READ
MSR_QTY	S12.6	CI_BSEG_READ
USAGE_FLG	A2	CI_BSEG_READ
USE_PCT	N3	CI_BSEG_READ
HOW_TO_USE_FLG	A2	CI_BSEG_READ

<b>Field Name</b>	<b>Format</b>	<b>Source/Value/Description</b>
CONSUM_SUB_FLG	A2	CI_REG
MSR_PEAK_QTY_SW	A1	CI_BSEG_READ
REGISTER_CONST	S6.6	CI_BSEG_READ
QTY_DEC_POS	N1	CI_UOM. Decimal positions of the final quantity.
FINAL_UOM	A4	CI_BSEG_READ
FINAL_TOU	A8	CI_BSEG_READ
FINAL_SQL	A8	CI_BSEG_READ
FINAL_REG_QTY	S12.6	CI_BSEG_READ

#### Bill Segment SQ (0900) Record Definition

<b>Field Name</b>	<b>Format</b>	<b>Source/Value/Description</b>
UOM_CD	A4	CI_BSEG_SQ
TOU_CD	A8	CI_BSEG_SQ
SQI_CD	A8	CI_BSEG_SQ
SQ_DEC_POS	N1	CI_BSEG_SQ
BILL_SQ	S12.6	CI_BSEG_SQ

#### End of Bill Segment (1000) Record Definition

<b>Field Name</b>	<b>Format</b>	<b>Source/Value/Description</b>
Dummy field		

#### End of Quote Detail (1100) Record Definition

<b>Field Name</b>	<b>Format</b>	<b>Source/Value/Description</b>
Dummy field		

#### End of Quote (9999) Record Definition

<b>Field Name</b>	<b>Format</b>	<b>Source/Value/Description</b>
Dummy field		

#### Address Sub-record Definition

The address that appears in both the quote record (0100) and the quote detail record (0200) is composed of the following:

<b>Field Name</b>	<b>Format</b>	<b>Description</b>
COUNTRY	A3	
ADDRESS1	A64	Refer to <a href="#">Completing A Quote Creates Quote Routings</a> for a description of conditions when this field will contain a fax number or an Email address rather than the first line of a postal address (this only occurs for a bill record (0100)).
ADDRESS2	A64	
ADDRESS3	A64	

Field Name	Format	Description
ADDRESS4	A64	
CITY	A30	
NUM1	A6	
NUM2	A4	
HOUSE_TYPE	A2	
COUNTY	A30	
STATE	A6	
POSTAL	A12	
GEO_CODE	A11	
IN_CITY_LIMIT	A1	

## Maintaining Quotes

A quote allows you to show a customer how their bill segments will look if a given set of pricing terms is applied to a given amount of consumption. If the customer accepts the quote, real service agreements are created.

Refer to [The Big Picture Of Quotations](#) for background information about quotes.

The topics in this section describe how to maintain quotes.

### Contents

- [The Lifecycle Of A Quote And Quote Detail](#)
- [Quote - Main](#)
- [Quote - Quote Details](#)
- [Quote - Quote Routings](#)

## The Lifecycle Of A Quote And Quote Detail

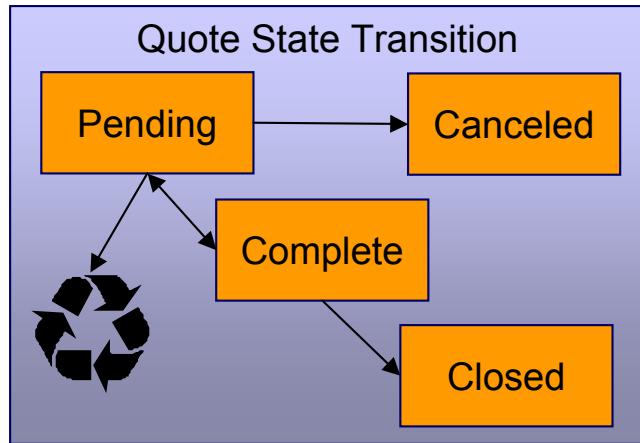
The topics in this section describe the lifecycle of the various quote objects.

### Contents

- [Quote Lifecycle](#)
- [Quote Detail Lifecycle](#)

### Quote Lifecycle

The following diagram shows the possible lifecycle of a quote.

**Pending**

A quote starts its life in the **Pending** state. Refer to [Creating Quotes And Quote Details](#) for more information.

A quote remains **Pending** until you **Cancel** or **Complete** it.

You may delete a **Pending** quote. Deleting a pending quote also deletes its quote details.

**Canceled**

You cancel a quote if you want a record of a **Pending** quote that was never sent to a customer.

**Complete**

A quote becomes **Complete** when it is ready to be sent to a customer. No changes may be made to a **Complete** quote. Refer to [Completing A Quote](#) for a description of what happens when a quote is completed.

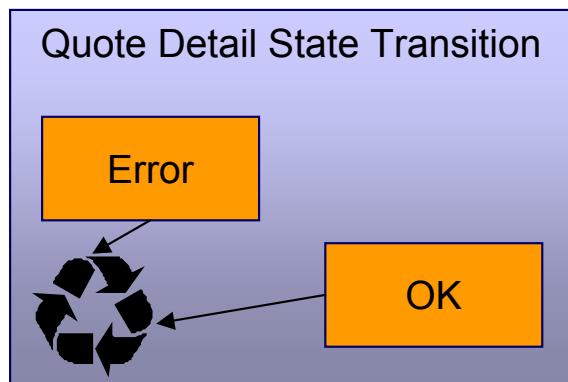
A **Complete** quote may be reopened. You would only do this if you need to prevent blatant errors from being sent to the customer. For example, if you notice an incorrect quote detail, you could reopen the quote, regenerate or delete the erroneous quote details, and then recomplete the quote.

**Closed**

A **Complete** quote becomes **Closed** when its terms are no longer valid. Refer to [Closing Quotes](#) for more information.

**Quote Detail Lifecycle**

The following diagram shows the possible lifecycle of a quote detail.



When a quote detail is generated, it is placed in the **Error** or **Ok** state depending on whether the system successfully generated simulated bill segments for each billing scenario linked to the related proposal SA. A quote detail may be deleted if its parent quote is **Pending**. Refer to [Creating Quotes And Quote Details](#) for more information.

## Quote - Main

This page is used to maintain quotes. This page provides many diverse functions:

- You can [generate](#), [complete](#), [close](#), reopen, and cancel a quote.
  - You can regenerate and delete quote details.
  - You can [accept](#), [decline](#) and cancel the proposal SAs that contribute a quote detail to the quote.

You should not attempt to use this transaction until you are comfortable with the concepts described under [The Big Picture of Quotations](#).

Open this page using **Sales & Marketing, Quotes**.

## Description of Page

**Quote Info** contains a concatenation of important information about the quote (e.g., its expiration date and status). **Quote ID** is the system-assigned unique identifier of the quote. These values only appear after the quote is added to the database.

**Account ID** identifies the customer or prospect associated with the quote. The name of the account's main customer appears adjacent.

**Quote Status** contains the state of the quote. Refer to [Quote Lifecycle](#) for information about the valid values and the state transition rules.

If your system has been configured to display an online image of a quote, a **Display Quote** button appears adjacent to the quote's status. When clicked, an online image of the quote is displayed. Refer to [Technical Implementation Of Online Quote Production](#) for how to enable this feature.

**Expiration Date / Time** identifies when the quote's terms are no longer valid. Refer to [Closing Quotes](#) for how the system uses this information to automatically close a quote.

Refer to [Quote Actions](#) for a description of each **Quote Action** button.

This transaction has sophisticated logic that can be used to perform mass updates to the quote's quote details and their related proposal SAs. These mass update actions are executed by selecting quoted details (in the grid) and then clicking the desired action button at the bottom the page. The remainder of this section describes how to use this mass update functionality.

The **Select All / Clear All** buttons are used to select / unselect the quote details. These buttons are disabled when the quote is **Closed**.

**100 quote details at a time.** Clicking **Select All** selects the first 100 quote details in the grid. If more than 100 quote details exist, you must select them in batches.

The grid that follows contains the quote details linked to the quote. The following information appears in the grid:

- Select box. Before you can perform any of the mass update actions at the bottom of the page, at least one quote detail must be selected. This field is disabled when the quote is **Closed**. You can use the **Select All** and **Clear All** buttons to change the value of the switch for all quote details in the grid.
- Click the go to button to transfer to the [Quote – Quote Detail](#) tab with the corresponding quote detail positioned in the scroll. On this page, you can see the details of the quote detail's simulated bill segments (and, from there, drill down into the details of each bill segment).
- The **Quote Detail Info** column contains a concatenation of a quote detail's division, SA type, start option, status, start date, and premise information (this information comes from the quote detail's proposal SA).
- The **Total Amount** column contains the sum of the quote detail's simulated bill segments.
- The **Remarks** column highlights special circumstances associated with the quote detail.
  - **Proposal SA has been changed since the quote was created** appears if the proposal SA was modified after the quote detail was generated. If you'd like to recalculate this quote detail, select this row and click the **Regenerate** button at the bottom of the page to regenerate the quote detail's simulated bill segments.
  - **Quote detail contains bill segments in error** appears if any of the simulated bill segments linked to the quote detail are in **error**. Click the adjacent go to button to transfer to the [Quote – Quote Detail](#) tab where the state of each simulated bill segment is shown. You can correct simulated bill segments that are in **error** by **Regenerating** the quote detail (and you do this by selecting the quote detail and clicking the Regenerate button at the bottom of the page).

The **Quote Detail Actions** and **Proposal SA Actions** can be used to perform mass updates to the selected quote details in the grid. Refer to [Quote Detail Actions](#) and [Proposal SA Actions](#) for the details.

## Action Buttons On Quote Main

The topics in this section describe the various action buttons on the [Quote – Main](#) page.

### Contents

- [Quote Actions](#)
- [Quote Detail Actions](#)
- [Proposal SA Actions](#)

## Quote Actions

The topics in this section describe the various **Quote Action** buttons on the [Quote – Main](#) page.

### Contents

- [Generate](#)
- [Complete](#)
- [Close](#)
- [Reopen](#)
- [Cancel \(Quote\)](#)

#### Generate

The **Generate** button causes one or more quote details to be created for the account's **Quotable** proposal SAs. This button is enabled when there is at least one **Quotable** proposal SA linked to the selected **Account ID**.

When clicked, the **Generate Quote** window opens.

You must specify the following parameters to generate quote details:

- Indicate if quote details should be created for **All Proposal SA's** or for a **Specific Proposal SA**.
- If you indicated a quote detail should be generated for a Specific Proposal SA, also specify the **Proposal SA ID**.

After specifying the parameters, click **Generate** to create the quote details for **quotable** proposal SAs.

Refer to [Creating Quotes And Quote Details](#) for more information.

**Warning!** If the quote has existing quote details for the designated proposal SAs, they are deleted when you generate quote details (i.e., the system regenerates existing quote details). You can also regenerate quote details by selecting them on the Main page and clicking the **Regenerate** button.

#### Complete

The **Complete** button causes the quote to become **complete**. This button is enabled when the quote is **pending** and all quote details linked to the quote are **OK**.

Refer to [Completing A Quote Creates Quote Routings](#) for a description of what happens during quote completion.

#### Close

A quote is closed when its terms are no longer valid (i.e., when the related proposal SAs can no longer be accepted). Most quotes are closed automatically by the system (refer to [Closing Quotes](#) for how this takes place).

If you want to close a quote real time, click the **Close** button. When this button is clicked, the system opens the **Close Quote** window and asks if anything should be done with the related proposal SAs.

In the resulting window, you must specify one of the following **Proposal SA Actions**:

- Choose **No Action** if you don't want to change the status of the related proposal SAs.
- Choose **Decline** if you want the status of the proposal SAs that have contributed quote details to the quote to be changed to **declined**. You must also define a **Decline Reason**.
- Choose **Cancel** if you want the status of the proposal SAs that have contributed quote details to the quote to be changed to **canceled**.

#### Reopen

The **Reopen** button causes a quote to return to the **pending** state. You would only reopen a quote if you need to prevent blatant errors from being sent to the customer. For example, if you notice an incorrect quote detail, you could reopen the quote, regenerate or delete the erroneous quote detail, and then recomplete the quote.

This button is enabled if the quote is **complete**.

#### Cancel (Quote)

The **Cancel** button causes the quote to become canceled. You would cancel a quote if you want a record of a **pending** quote that was never sent to a customer. This button is enabled when the quote is **pending**.

Canceling the quote does NOT cancel the related proposal SAs. If you want to cancel the related proposal SAs, you have the following options:

- You can click the **Close** button rather than the **Cancel** button. When you close a quote, you are given the option of also canceling the proposal SAs that contributed a quote detail to the quote.
- Before canceling the quote, you can select the quote details and click the **Cancel** button at the bottom of the page (in the Proposal SA Actions section). When you click this button, the system cancels the proposal SAs linked to the quote details.
- You can transfer to the [Quote - Quote Details](#) tab and then drill down to the each proposal SA (using the SA context menu button). This transfers you to [Service Agreement - Main](#) where you can cancel the proposal SA.

#### Quote Detail Actions

The topics in this section describe the various **Quote Detail Action** buttons on the [Quote – Main](#) page. These buttons are used to perform a given action to the quote details that are selected in the quote details grid.

#### Contents

- [Regenerate](#)
- [Delete](#)

#### Regenerate

The **Regenerate** button is used to delete and recreate the selected quote details. This button is enabled if the quote is **pending** and at least one row in the quote details grid is selected.

#### Delete

The **Delete** button is used to delete the selected quote details. This button is enabled if the quote is **pending** and at least one row in the quote details grid is selected.

## Proposal SA Actions

The topics in this section describe the various **Proposal SA Action** buttons on the [Quote – Main](#) page. These buttons are used to perform a given action to the proposal SAs associated with the selected quote details.

### Contents

- [Accept](#)
- [Decline](#)
- [Cancel](#)

#### Accept

The **Accept** button is used to accept the proposal SA associated with the quote detail. This button is enabled if the quote is ***pending*** or ***complete*** and at least one row in the quote details grid is selected.

When you accept a proposal SA, you are asked to supply the **Start Date** of the real service agreement that will be created.

Refer to [Accepting / Declining Quote Details](#) for more information about how accepting a quote detail causes a real service agreement to be created.

#### Decline

The **Decline** button is used to decline the proposal SA associated with the quote detail. This button is enabled if the quote is ***pending*** or ***complete*** and at least one row in the quote details grid is selected. When you decline a proposal SA, you are asked to supply a **Decline Reason**.

Refer to [Accepting / Declining Quote Details](#) for more information about declining a quote detail.

#### Cancel

The **Cancel** button is used to cancel the proposal SA associated with the quote detail. This button is enabled if the quote is ***pending*** or ***complete*** and at least one row in the quote details grid is selected.

## Quote - Quote Details

This page is used to view a summary of the simulated bill segments linked to a quote detail.

Open this page using **Sales & Marketing, Quote** and navigate to the **Quote Details** tab to define this information. You can also display a quote detail by clicking the respective go to button in the quote details grid on the main tab.

### Description of Page

**Quote Info** contains a concatenation of important information about the quote (e.g., its expiration date and status). **Quote ID** is the system-assigned unique identifier of the quote. These values only appear after the quote is added to the database.

The **Quote Details** scroll contains one entry for every quote detail linked to the quote (a summary of these quote details also appears on the Main tab). The remainder of this section describes the information displayed for each quote detail.

**You don't have to use the scroll.** If the quote has many quote details, you don't have to use the scroll to work through them. Rather, you can use the go to button in the quote details grid on the main tab to position the respective quote detail in the following scroll. Also note, you can use Alt – and Alt + to scroll to the next and previous quote details in the scroll.

**You cannot maintain quote details on this page.** All maintenance functions (generate, delete, accept, decline and cancel) are performed on the **Main** tab.

**Premise Info** is the address of the main premise associated with the service agreement (as defined on the service agreement's characteristic premise). It's important to note that the service agreement may cover service at multiple premises, but only one may be defined as the main premise.

**Quote Detail ID** is the system-assigned unique identifier of the quote detail.

**SA Info** is a concatenation of the proposal SA's division, SA type, start option (if any), status and start date.

**SA Version** is a snapshot of the proposal SA's version number at the time the quote detail was generated. This information is used by the system to detect if the proposal SA was modified after the quote detail was created.

**Total Amount** is the sum of the quote detail's simulated bill segments.

**Create Date / Time** is the date and time the quote detail was generated.

**Remarks** highlight special circumstances associated with the quote detail. Please see the description of this field on the main tab for more information.

The grid contains a row for each simulated bill segment linked to the quote detail. The following information appears in the grid:

- The first go to button transfers you to the [bill segment page](#) where the details of the simulated bill segment are shown.
- **Bill Segment Period** defines the start and end dates of the simulated bill segment.
- **Amount** contains the amount of the bill segment.
- **Remarks** highlight special circumstances associated with the bill segment.
- **Billing Scenario ID** allows you to go to the proposal [SA's billing scenario](#) where you can tinker with consumption and the bill segment period used to generate this bill segment. Refer to [Proposal SAs Contain Billing Scenarios](#) for more information.

## Quote - Quote Routings

This page is used to view the recipients of a quote. Refer to [Printing Quotes](#) for a discussion of how this information is used to produce quote images.

This page is also used to request a new copy of a quote.

**No routing information until completion.** A quote has no routing information until it is **complete**. At [completion time](#), the system creates a routing detail for every person linked to the account that receives a copy of its quotes. Refer to [Printing Quotes](#) for more information about how this information is constructed.

Use **Sales & Marketing, Quote, Quote Routings** to open this page.

### Description of Page

**Quote Info** contains a concatenation of important information about the quote (e.g., its expiration date and status). **Quote ID** is the system-assigned unique identifier of the quote. These values only appear after the quote is added to the database.

The **Quote Routing Information** scroll contains an entry for each recipient of the quote. This information is disabled after the quote has been routed to the recipient. The remainder of this discussion describes the fields that can be defined for each recipient of the quote.

**Person ID** identifies the person who was originally associated with the routing record. You may change the **Person ID** to any person in the system as this information is only used to default the recipient's name and address information, below.

The **Quote Routing Parameters** area contains information explaining how and when the quote is routed to the recipient. This information is gray when the quote routing information has been extracted or if a quote is reopened:

- **Quote Route Type** is the method used to transmit the quote to the customer. This information defaults from [Account – Person Information](#).
- **Batch Control** is the identifier of the batch of quotes in which the recipient's copy was downloaded. Embedded in this identifier is the batch control ID and the batch number. Note, this batch process is defined on the **Quote Route Type**.
- **Sequence** is the system-assigned identifier assigned to this quote routing.
- Turn on **Do Not Extract** if the quote should not be routed to the individual for whatever reason.
- **Reprint** is checked if an operator created this routing. This switch is populated by the system and is always protected.
- **Extract Date/Time** is the date and time the quote was downloaded. This information only appears after the download has commenced.

The **Mailing Address Information** area contains the recipient's name and address.

- If the recipient has an [override mailing name](#), **Name1**, **Name2** and **Name3** contain the person's override mailing name. Otherwise, **Name1** contains the recipient's primary name (this name might also include a prefix or suffix if the related [Account – Person](#) relationship has these fields defined).
- The remaining fields are the recipient's address. The information that's displayed is dependent on the [Quote Route Type](#).
  - If the recipient's **Quote Route Type** indicates a **postal** routing method is used, the address constituents are populated with the address specified on the **bill** address source that is also specified on the [account / person information](#). Note that the **Country** associated with the address controls the fields that appear in the address. Refer to [Defining Countries](#) for more information on the address constituents.

- If the recipient's **Quote Route Type** indicates quotes are routed via fax, the recipient's **Fax ID** appears. The system knows which of a person's phone numbers is a fax number by the phone type.
- If the recipient's **Quote Route Type** indicates quotes are routed via email, the recipient's **Email Address** appears.
- Please note the following in respect of the **Mailing Address Information**:
  - If the recipient's address is a mailing address (as opposed to fax or email), the **Country** associated with the address controls the fields that appear in the address. Refer to [Defining Countries](#) for more information on the address constituents.
  - If a user manually adds a routing record, the mailing address information defaults from the information on the current routing record.
  - If a user specifies a **Person ID** that is linked to the quote's account, the **Quote Route Type** defined for this person on the account is defaulted.
  - If a user specifies a **Person ID** that is not linked to the quote's account, the **Quote Route Type** on the [installation record](#) is defaulted. Note, if the quote route indicates the quote is routed via the postal service, the system will only be able to default the person's address if the person has an override mailing address on their [correspondence information](#).
  - If the user changes the **Quote Route Type** after the initial default, the system populates the address information as described above.

# Service Credits

“Service credits” is a term used to describe a special program designed to reward a customer for service along with benefits related to such a program. For each program that a customer participates in, there is a service credit membership record. Service credit events are used to record the “points” earned and redeemed for the membership. The topics in this section provide further details related to the service credit functionality.

For information about designing and setting up your control tables for configuring service credit functionality, refer to [Designing Your Service Credit Options](#).

**Service credits are optional.** Service credits may not be set up in your system; they are only used if your company offers rewards programs to customers.

**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.

## Contents

- [The Big Picture Of Service Credit Membership](#)
- [The Big Picture of SC Events](#)
- [Maintaining Service Credit Memberships and Their Events](#)

## The Big Picture Of Service Credit Membership

If a customer is participating in a rewards program, the customer has a service credit membership. The topics in this section provide background information about a variety of service credit membership features and functions.

## Contents

- [Who are the Members?](#)
- [Membership Application](#)
- [Creation of a Membership](#)
- [Membership Fees](#)
- [Stopping a Membership vs. Inactivating a Membership](#)

## Who are the Members?

Each membership has one or more “members”. The members are defined by the accounts linked to the membership and by the financially responsible person(s) linked to each account.

A typical membership is linked to only one account. However, in some cases more than one account is linked to the membership. For example, a customer may have one account for residential service and one for commercial service, but both accounts are part of the same membership. In this example, one person with multiple accounts is linked to the membership, so there is only one member.

For each account linked to the membership, the service related to the account may be used to calculate the accumulated credits for the membership. Refer to [How Are Service Credits Earned?](#) for more information.

**Note.** An account may not be linked to more than one membership of the SAME membership type at a given point in time. In other words, an account may not have two United Airlines frequent flier memberships (otherwise, the account would accumulate double miles). However, an account may be linked to more than one membership if they are of different membership types. For example, an account may have a capital credits membership and a United Airlines frequent flier membership.

When you find a customer's record on [Control Central - Main](#), you are transferred to the [Account Information](#) portal. One of the zones on this page contains information about memberships for the customer.

**Portal Preferences.** The service credit membership zone only appears if you have modified your preferences to display this zone. Refer to [Portal Preferences](#) for more information.

## Contents

[Joint Membership](#)

[Changing the Membership Accounts](#)

## Joint Membership

It is possible for more than one person to belong to the same membership. These persons may be linked to the same account or to different accounts. A typical scenario would be a married couple that shares in a membership. This is considered a "joint" membership.

The number of "members" for a membership is mainly for information purposes and typically has no bearing on the service credit amounts calculated for the membership. There may be legal restrictions or business practice restrictions controlling how you may set up your members. For example:

- In an airline's frequent flier program, a separate frequent flier account would exist for each customer. It is possible for you to create a frequent flier membership with more than one "member", but for the system, the important information is the external frequent flier miles number linked to the membership. The customer linked to this number accumulates the miles. For this case, your business practice may be that the person with the frequent flier account must be the only financially responsible person.
- For a capital credits membership, you may be required to distinguish between "single" memberships and "joint" memberships. If a single person gets married and would like the spouse to be added as a member, your practice may be to expire the single membership and to create a new "joint" membership to include both spouses.

**Co-op Members.** Do not confuse the “members” of a co-op with the members linked to a service credit membership. At a cooperative, most customers are considered “members” of the co-op. However, in the system, separate membership records are created for each single or joint membership.

## Changing the Membership Accounts

The link between the accounts and the membership is NOT effective dated. This means that if the list of accounts for a membership changes, you need to stop the membership and start a new one with the new list of accounts.

**Stopping Memberships.** Stopping a membership does not necessarily prevent new service credits from accumulating. Refer to [Stopping a Membership vs. Inactivating a Membership](#) for more information.

## Membership Application

Your business practice may dictate that the customer complete an enrollment form. Depending on your practice, the form may need to be received prior to starting membership or it may simply be a mechanism to gather supplemental information once the membership has begun.

Either way, you may consider setting up the system to handle forms as follows:

- Design your membership type to include a creation algorithm that generates a letter when the membership is created. The letter template could include the form to be filled out.
- Create the new membership in ***pending*** status. The letter and form are sent to the customer and the pending status indicates that you are waiting for the completed form. Refer to the [Lifecycle of a Membership](#) for more information.
- When the paperwork is received, change the status of the membership to ***active***.

**Membership Events for Pending Memberships.** The system allows events to be created for ***pending*** memberships assuming that paperwork is supplemental information that should not prevent service credits from accumulating. If your business practice dictates that the paperwork must be received prior to accumulating service credits, the algorithm or background process that creates events for memberships of this type should be coded to only create events for ***active*** memberships. Refer to [How Events Are Created](#) for more information.

## Creation of a Membership

Memberships may be created using the [membership](#) page when a customer has signed up for a special program. In addition, it may be your business practice to automatically create a membership under certain circumstances when a customer has signed up for service.

For example, if you are a co-op, perhaps every new customer is automatically enrolled in a capital credits membership. Or perhaps signing up for cable service automatically entitles your customers to membership in the free pay-per-view movies program. For these cases, the recommendation is to use the [order](#) page to start service for your customer and to use questions and miscellaneous fields to record information needed to set up the new membership correctly. Refer to [Using Campaigns & Packages to Create a Membership](#).

## Membership Fees

Some memberships may require a fee to be paid by the customer when joining. The behavior of this fee may differ depending on whether it is refundable or not.

- If the fee is nonrefundable, you create an SA with a simple adjustment or a one time billable charge to levy the fee. Refer to [Defining Control Tables for a Nonrefundable Fee](#).
- If the fee is refundable, in addition to defining the SA to hold the fee, you need to put code in place to initiate the refund. Refer to [Setting Up a Refundable Fee](#) for more information.

## Stopping a Membership vs. Inactivating a Membership

You may stop a membership by entering an end date for the membership. However, depending on how events are created for your membership, stopping the membership may not prevent new events from being created. For example:

- If you stop a frequent flier membership on May 15<sup>th</sup> and a bill is subsequently created for the period May 1<sup>st</sup> through June 1<sup>st</sup>, your business rules may dictate that frequent flier miles should be accumulated for this bill because the membership was active for part of the period covered by the bill.
- For a capital credits membership, capital credit allocations are calculated for the prior year. If a membership stopped in the middle of the previous year, the membership should still get capital credits for the part of the year that the membership was still in effect.
- Capital credit retirements are calculated based on capital credit allocations accumulated in the past. If a membership has been expired, the members are still eligible for retirement amounts for the accumulated capital credit allocations. These retirements may continue for many years after a membership is stopped.

**Membership Status.** For all the above scenarios, the system assumes that the status of the membership remains **active**.

You may have some circumstances where you do not want new events to accumulate after a membership is stopped. To prevent new events from being created, you must change the status of the membership to **inactive**.

If a customer moves away or otherwise stops service with your company, your business rules dictate whether or not the membership should be stopped or changed to **inactive**. The following issues must be considered:

- Will service credit events still get created for these memberships? If so, is that acceptable?
  - For frequent flier memberships, the miles are calculated based on new bill amounts for the membership's service agreements. If the SAs that contribute to the membership are expired, no new bills are generated so no new miles are accumulated.

- For capital credits, after a customer moves away, the capital credit allocation for the final year of the customer's service may still need to be calculated. In addition, this customer is eligible for the capital credit retirement amounts, even though they are no longer a customer.
- What happens if the customer returns to your company and signs on for service again? Assuming that the same account number is used, is it ok for the membership to be automatically reinstated? Or should the customer reapply? If the customer should reapply, you may consider updating the membership with a stop date when the customer originally stopped service.

**Note.** There is nothing in the system logic that automatically updates the membership stop date or the status when service agreements that contribute to the membership are stopped.

## The Big Picture of SC Events

The service credit events record the accumulation and redemption of awards earned by the membership. The topics in this section provide background information about a variety of service credit event features and functions.

**Events.** For convenience, this document often refers to service credit events simply as "events".

### Contents

- [Event Amounts May Contribute to a Balance](#)
- [Events May Indicate a Fiscal Year](#)
- [Events May Indicate a Subcategory](#)
- [An Event May Cause Other Actions to Occur](#)
- [Financial Transactions for an Event](#)
- [How Events Are Created](#)
- [Creating Pending Events](#)
- [Events May Be Transferred](#)
- [CPCRALOC - Allocating Capital Credits](#)
- [CPCRRETR - Capital Credit Retirement](#)

## Event Amounts May Contribute to a Balance

The event amount may be positive or negative, based on whether the event is accumulating points earned for the membership or redeeming points for the membership. For some types of memberships, these credit and debit amounts should be added together to produce the overall balance for the membership.

For some types of memberships, a balance is not applicable. This is true for memberships where the redemption of the points does not occur within the system. For example, if the membership accumulates frequent flier miles on behalf of an airline, the customer redeems the earned miles with the airline, not with your company. As a result, there is no need to calculate a balance for the membership.

If you want the system to calculate a balance, you must set a switch on the [membership type](#) page. For memberships that calculate a balance, the balance may be viewed on the membership page.

## Events May Indicate a Fiscal Year

For some types of memberships, the amount of each event is associated with a specific fiscal year. This is true for capital credit memberships where capital credit allocations are calculated for a fiscal year. When the capital credits are redeemed (also called retired), the retirement events must also indicate a fiscal year. Because all events for this type of membership are related to a fiscal year, the system can calculate the balance for the events by fiscal year.

If you want the system to require a fiscal year for every event in the membership, you must set a switch on the [membership type](#) page. For memberships that calculate a balance and indicate a fiscal year, the balance for each fiscal year may be viewed on the membership page.

## Events May Indicate a Subcategory

For some types of memberships, the amount of each event is further grouped by a subcategory. This is common for capital credit memberships where multiple capital credit allocations are calculated. For example, each year, you may calculate allocations for generation and for transmission. Events for memberships of this type must indicate a subcategory.

Cooperatives that allocate capital credits for more than one subcategory, typically want to view the balance of the events over time broken down into these subcategories. When the capital credits are redeemed or retired, the retirement events must also indicate a subcategory. Because all events for this type of membership are related to a subcategory, the system can calculate the balance for the events by subcategory.

**Partial Retirement.** For some cooperatives, it is common that certain capital credit allocations are never retired. Refer to [Partial Retirement](#) for more information.

If you want the system to require a subcategory for every event in the membership, you must define the valid subcategories on the [membership type](#) page. For memberships that calculate a balance and indicate a subcategory, the balance for each subcategory may be viewed on the membership page.

**Calculating Subcategory Balances.** If events for your membership indicate a subcategory and fiscal year, balances for the membership are calculated for both groupings.

## An Event May Cause Other Actions to Occur

When an event is generated, there are often other actions that need to take place. For example:

- If an event is created that allocates points for the customer, you may want to generate a temporary message on the customer's next bill.
- If a capital credit allocation event is created, the general ledger is affected so an adjustment must be created to post to the general ledger. Refer to [Financial Transactions for an Event](#) for more information.
- If an event is created to add frequent flier miles, the event must be marked for extraction so that the information is sent to the appropriate airline. Refer to [Interface Membership Information to a Third Party](#) for more information.

- If an event to retire capital credits is created, the customer's balance is affected so an adjustment must be created to transfer the capital credit to the members' service agreements. Refer to [Financial Transactions for an Event](#) for more information.

The system relies on algorithms linked to the event type to perform the actions described above and any other action that may be required by your business.

Refer to [Service Credit Event Types](#) for more information about the algorithms provided with the base product.

**Event Cancellation Algorithms.** The event type also defines algorithms to be executed when an event is canceled.

## Financial Transactions for an Event

An event may be related to financial transactions. There are two types of financial transactions that may be linked to the event:

- Financial transactions that **contributed to** the event. These are bills, payments or adjustments that caused the event to be created. For example, if you generate an event for frequent flier miles as a result of some amount billed for the customer, the bill segments associated with the billed amount are linked to the event. The FTs are linked to the event for audit purposes.
- Financial transactions that are **created by** the event. In [an event may cause other actions to occur](#), we saw examples where an event completion algorithm may cause adjustments to be created. The FT ID for these adjustments is linked to the event as audit.

**Automatically Canceling Event Financial Transaction.** The **created by** financial transactions may get canceled if the event is canceled AND you have plugged in an appropriate [Event Cancellation Algorithm](#).

For financial transactions that are created by the event, the system must post the financial transactions to one or more appropriate service agreements. The service agreement to use depends on your business rules. For example:

- If the event affects the balance of accounts linked to the membership, the event completion algorithm may create adjustments for [service agreements that contribute to the membership](#).
- If the event affects the general ledger and does not affect any balance for the account, the event completion algorithm simply needs a service agreement to use for the resulting adjustment. Rather than arbitrarily picking any service agreement in this case, a special [miscellaneous service agreement](#) should be linked directly to the membership for such adjustments.

## How Events Are Created

Although a user may create an event manually on the [service credit event](#) page, most events are created automatically by an algorithm or by a background process. The following topics further describe these options.

## Contents

- [Events Created by an Algorithm](#)
- [Events Created as a Result of a Background Process](#)

### Events Created by an Algorithm

You would design an algorithm to create your events if the event should be triggered by an occurrence in the system, such as starting service or generating a bill. For example, perhaps every time you are billed for electric and/or gas service, the system adds one frequent flier mile for every \$10 spent on service. Refer to [How Are Service Credits Earned?](#) for more information.

The base product provides a bill completion algorithm to create an event as a result of the amount billed for service agreements linked to the membership. Refer to [CBCM-SC](#) for more information.

### Events Created as a Result of a Background Process

You would use a background process to create the event if the calculation is done periodically or on an ad-hoc basis and does not get triggered by something occurring in the system. For example, the capital credit allocation and retirement processes are typically run once a year, when the business has determined that they are ready to calculate these amounts.

Refer to [Allocating Capital Credits](#) and [Capital Credit Retirement](#) for more information.

## Creating Pending Events

Events may be created in pending or complete status. Only complete events cause the completion event algorithms to run and [cause other actions to occur](#). You may use the status of **pending** if you want to review the calculated amount prior to completing the event.

For example, when a cooperative runs the background process to [allocate capital credits](#), it may create the new events in the **pending** status. This allows the co-op to review the total amount of capital credits calculated for all members. If this total is in line with the co-op's expectations, they run the background process again creating the new events in **complete** status. Only at this point do the completion event algorithms run.

## Events May Be Transferred

In some situations, you may want to transfer service credit events from one membership to another on the [service credit event](#) page. For example, if a member dies, you may decide to transfer earned service credits to the membership for a beneficiary. When transferring a service credit:

- The target membership must be of the same membership type as the original membership.
- The original service credit event is canceled and any event cancellation algorithms linked to the [SC event type](#) are executed.
- A new service credit event is created and completed and any event completion algorithms linked to the [SC event type](#) are executed.
- The new service credit event references the original event ID in the **Transferred From** event field.

**Note.** **Contributed to** FTs linked to the original event are not transferred to the new event. If you have **created by** FTs linked to the original event, the recommendation is to ensure that an event cancellation algorithm cancels this FT and that an event completion algorithm for the new event creates an appropriate new FT for the new membership.

## CPCRALOC - Allocating Capital Credits

The base product provides a background process that allocates capital credits to members of a cooperative. The result of this process is the creation of new service credit events for capital credit memberships. This background process is typically run after the profits for the previous year are calculated. During that time, this job could be run a number of times for any of the following reasons:

- At the beginning of the allocation process, you may know the total amount to allocate to all members; however, you may estimate the allocation factor that determines the individual amounts that get distributed to the members. You may need to change your estimate and rerun the process until all the individual allocations add up to the desired total amount. To run the job with the estimated allocation factors, indicate that you want events created as pending. When you confirm the correct allocation factor, a final run should be executed with an indication that the events should be created as complete. Every time the process runs, it will first remove any pending events for the same input event type, subcategory and fiscal year. Refer to [Creating Pending Events](#) for more information.
- If the cooperative has more than one subcategory, the process must be run for each subcategory. In fact, the above trial-and-error process may need to be followed for each subcategory.

You must provide the [service credit event type](#) to use for the new events. This process determines the membership type related to the event type. The process then finds all pending and active memberships for this membership type. To calculate the capital credit allocation, the system finds all applicable bill segments for each [service agreement that contributes to the membership](#).

Capital credit allocations are calculated for a specific [fiscal year](#). The fiscal year is input to the background process and is stamped on any new events created by the process. The system also uses the fiscal year to determine the applicable bill segments for the membership's SAs. It uses the bill segment's accounting date and the [accounting calendar](#) to determine the fiscal year for each bill segment.

**Canceled Bill Segments.** Any bill segments that were canceled in the fiscal year are included as well.

**Accounting Calendar.** This process assumes that only one accounting calendar exists for the company. This is the case for companies that operate in a single jurisdiction, which is true for cooperatives.

**Allocation Period.** If the membership was only effective for part of the fiscal year, the allocations are only calculated for the period of time that the membership was in effect.

For each applicable bill segment, the system determines the SQ total and the total sales amount for each segment. To calculate the totals correctly, a characteristic type and value to identify SQ bill lines must be provided to the background process and a characteristic type and value to identify bill lines contributing to the total sales amount must be provided to the background process.

Refer to [Identifying SQ and Sales Information for Historical Bill Segments](#) for more information.

If your membership type indicates subcategories to be used for the events, you must provide a subcategory as input. If you have multiple subcategories for your membership, this process must be run once for each subcategory.

The calculation of the capital credit allocation amount typically uses an allocation factor. This factor is often different for each subcategory. You will probably define a different bill factor for each subcategory and use the appropriate bill factor when submitting the process for each subcategory.

This process also assumes that the allocation factor differs by the customer's revenue class. It assumes that the bill factor's characteristic type is revenue class. Refer to [Designing Bill Factors for Credit Allocation](#) for more information. For each service agreement linked to the membership, the appropriate allocation factor is determined based on the revenue class for the SA type.

Capital credits for the input subcategory are calculated as follows:

$$\text{Fiscal year sales amount} - (\text{fiscal year SQ amount} * \text{revenue class allocation factor}) = \text{Allocation Amount}$$

The calculated amount is compared against a minimum amount, input to the background process. If the calculated amount is less than the minimum, no event is stored for this amount.

After determining the amount, a new service credit event is created for the membership. The event is stamped with the input subcategory, service credit event type and fiscal year.

The FTs for all the bill segments used to determine the allocation amount are linked to the event as **contributed to** FTs. Refer to [Financial Transactions for an Event](#) for more information.

**Rerunnable Process.** This process is restartable and rerunnable. When running the process, for each membership the system checks whether there is already a **complete** event for the input service credit event type, subcategory and fiscal year. If so, it does not create a new event.

Refer to [CPCRALOC](#) for more information.

## CPCRRETR - Capital Credit Retirement

**Calculation Method.** There are various calculations that different cooperatives may use when determining the amount to retire. The base product has provided one calculation method.

This background process is run when a cooperative is ready to pay out allocated capital credits to its members. This process is referred to as retirement. The result of this process is the creation of new service credit events for capital credit memberships. Capital credits are paid as the cooperative's financial position permits. During that time, this job could be run a number of times for any of the following reasons:

- You may choose to run the process and [create pending events](#) to review the results prior to finalizing the retirement. When you are happy with the results, a final run should be executed with an indication that the events should be created as complete. Every time the process runs, it will first remove any pending events for the same input event type and subcategory.
- If the cooperative has more than one subcategory, the process may be run separately for each subcategory. You may also run the process with no subcategory indicated and then all subcategories are processed.

**Partial Retirement.** For some cooperatives there may be capital credits for some allocated amounts that are never retired. If that is the case, simply run this process for each subcategory whose amounts do get retired.

You must provide the [service credit event type](#) to use for the new events. This process determines the membership type related to the event type. The process then finds all pending and active memberships for this membership type.

This process receives a percentage to retire as input. It determines the current balance of all completed events for the subcategory. The retirement amount for the subcategory is calculated as the current balance \* input percentage to retire. The process then determines the outstanding balance for each subcategory and fiscal year combination. For every combination that still has a balance, the process creates events to relieve the outstanding balance for the oldest fiscal year first. It creates events for subsequent fiscal years until the full retirement amount is used.

In order for the process to be rerunnable and restartable, a Control ID is used to mark retirement events that are created by the process. The Control ID is input to the routine and is determined by the user. When the process runs, it determines whether an event already exists for the subcategory / Control ID combination. If so, new events are not created for that subcategory. If you run the process multiple times for the reasons described above, you should use the same Control ID each time. This way the system knows that the process is being run again for the same retirement calculation.

Refer to [CPCRRETR](#) for more information.

## Maintaining Service Credit Memberships and Their Events

The topics in this section describe how to maintain a service credit membership and its events.

Refer to [The Big Picture Of Service Credits](#) for background information about service credits.

### Contents

## Service Credit Membership

### Maintaining Service Credit Events

# Service Credit Membership

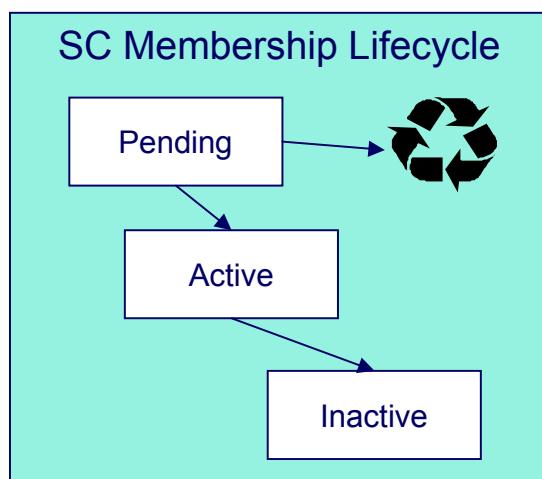
The topics in this section describe how to maintain service credit memberships. Refer to [The Big Picture Of Service Credit Membership](#) for background information about memberships.

## Contents

- [Lifecycle of a Membership](#)
- [Service Credit Membership - Main](#)
- [Service Credit Membership - Account](#)
- [Service Credit Membership - Characteristics](#)
- [Service Credit Membership - Membership Tree](#)

## Lifecycle of a Membership

The following diagram shows the possible lifecycle of a service credit membership.



### Pending

A service credit membership may start out in the **Pending** state. A membership is typically created in this state if the creation of the membership requires approval or any manual step (e.g. waiting for paperwork) that needs to be completed prior to activation.

**Events for Pending Memberships.** Service credit events may still be created for membership records in the **Pending** state.

### Active

A service credit membership is in the **Active** state when a customer is actively participating in a service credit membership. A membership may be created in the **Active** state or it may transition to the **Active** state from **Pending** when all the necessary approval or paperwork is complete.

**Manual Activation Only.** There is no system functionality that automatically changes the status of a membership from **Pending** to **Active**. This is a manual step.

### Inactive

If a customer is no longer actively participating in a service credit membership, change the status to **Inactive**. This will prevent any new events from being created for the membership.

## Service Credit Membership - Main

This page is used to maintain and display general information about a service credit membership. Open this page using **Customer Information, Service Credit Membership**.

### Description of Page

**Membership Info** and **Membership ID** are displayed on every page. These values only appear after the membership exists on the database. The **ID** is a system-assigned random number. The **Membership Info** is a concatenation of important details about the membership.

**Membership Status** defines the state of the membership.

Refer to [Lifecycle of a Membership](#) for more information about membership status.

Click the **Activate** button to activate a **Pending** membership. When this button is clicked, the system executes any Membership Activation algorithms that are linked to the [service credit membership type](#).

Click the **Inactivate** button to transition an **Active** membership to **Inactive**. When this button is clicked, you are prompted to provide an **Inactive Reason** and the system executes any Membership Inactivation algorithms that are linked to the [service credit membership type](#).

Indicate the **SC Membership Type**. This field is important as it controls numerous aspects of the behavior of the membership and its events.

For more information about what the membership type controls, refer to [Setting Up Service Credit Membership Types](#).

Indicate the **Start Date** and **End Date**, which define the period for which this membership is effective. Leave the end date blank if this membership is effective indefinitely. Refer to [Stopping a Membership vs. Inactivating a Membership](#) for more information.

If this membership is related to a record in an external system, use the **External ID** to indicate the unique identifier of the record in the external system.

If events linked to your membership may generate miscellaneous adjustments, indicate the **Service Agreement** to use for these [miscellaneous transactions](#). The account for this service agreement must be one of the accounts linked to the membership.

The **Events** grid displays the events linked to the membership. To view any of the events use the drill down button.

If your SCM type indicates that the membership has a balance, the tree at the bottom displays information about the balance of the membership based on the amounts for **completed** events. If the SCM Type indicates subcategories and/or a fiscal year is required, subtotals for **completed** events are also displayed for each subcategory / fiscal year and by fiscal year / subcategory, if applicable. If any **pending** events exist, the tree includes a node with information about these events.

## Service Credit Membership - Account

This page is used to maintain the accounts linked to a service credit membership. Open this page using **Customer Information**, **Service Credit Membership** and navigate to the **Account** tab.

### Description of Page

Indicate the **Account ID** for each account linked to this membership. Typically a membership will be linked to only one account. However, if a person has multiple accounts and the service credits should span the accounts, multiple accounts are linked to the same membership.

The Service Credit Membership zone on the Account Information portal contains information about memberships for the current customer. (When you find one of the customer's records on [Control Central - Main](#), you are automatically transferred to the [Account Information](#) portal.)

**Portal Preferences.** The Service Credit Membership zone only appears if you have modified your preferences to display this zone.

## Service Credit Membership - Characteristics

The characteristics page contains information that describes miscellaneous information about the service credit membership. Open this page using **Customer Information**, **Service Credit Membership** and navigate to the **Characteristics** tab.

### Description of Page

**Characteristic Types.** You can only choose characteristic types defined as permissible on the service credit membership record. Refer to [Setting Up Characteristic Types & Their Values](#) for more information.

To modify a characteristic, simply move to a field and change its value. To add a new characteristic, click + to insert a row then fill in the information for each field. The following fields display:

#### Effective Date

Define the date on which the characteristic becomes effective. The effective date defaults from the [Installation Record](#).

#### Characteristic Type

Indicate the type of characteristic.

#### Characteristic Value

Indicate the value of the characteristic.

**Default Note.** A service credit membership's characteristics default from the [service credit membership type](#).

## Service Credit Membership - Membership Tree

The membership tree page displays accounts, SAs and persons linked to the service credit membership. Open this page using **Customer Information, Service Credit Membership** and navigate to the **Membership Tree** tab.

### Description of Page

The tree displays each account linked to the membership. For each account, a node exists for:

- Each service agreement whose SA type is included in the service credit [membership type](#)'s SA type collection.
- Each financially responsible person linked to the account. These persons are considered the [members](#).

## Maintaining Service Credit Events

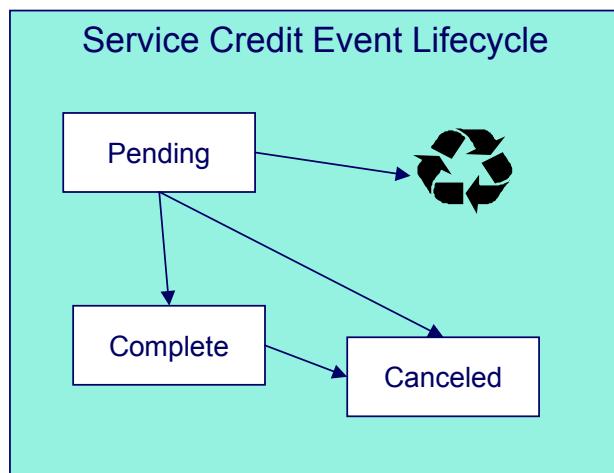
Service credit events record credits earned or redeemed for a given membership. The topics in this section describe how to maintain service credit events. Refer to [The Big Picture of Service Credit Events](#) for background information about memberships and their events.

### Contents

- [Lifecycle of a Service Credit Event](#)
- [Service Credit Event - Main](#)

### Lifecycle of a Service Credit Event

The following diagram shows the possible lifecycle of a service credit event.



#### Pending

A service credit event may start out in the **Pending** state. An event is typically created in this state when a review of the event is desired before completing it. Refer to [Creating Pending Events](#) for more information.

**Complete**

A service credit event is typically created in the **Complete** state. When an event is completed, the system executes any event completion algorithms indicated on the [service credit event type](#).

**Canceled**

If a service credit event is no longer valid, it may be **Canceled**. When an event is canceled, the system executes any event cancellation algorithms indicated on the [service credit event type](#).

## Service Credit Event - Main

Open this page using **Customer Information, Service Credit Event**.

### Description of Page

The **Service Credit Event Info** is a concatenation of important details about the service credit event. The **Service Credit Event ID** is a unique system-assigned random number. These values only appear after the service credit event exists on the database.

**Service Credit Event Status** defines the state of the service credit event.

Refer to [Lifecycle of a Service Credit Event](#) for more information.

Click the **Complete** button to change a **Pending** event to **Complete**. When this button is clicked, the system executes any event completion algorithms that are linked to the [service credit event type](#).

Click the **Transfer** button to transfer this event to another membership. When this button is clicked, you are prompted to provide the membership ID to transfer the event to. The new membership must have the same membership type as the original membership. Refer to [Events May Be Transferred](#) for more information.

Click the **Cancel** button to cancel this event. When this button is clicked, the system executes any event cancellation algorithms that are linked to the [service credit event type](#).

The **Create Date/Time** captures the date and time that this event was created. This defaults to the system date/time.

Indicate the service credit **Membership ID** to which this event should be linked.

Indicate the **Service Credit Event Type**. This field may define algorithms to be executed when the event is completed or canceled. Refer to [Setting Up Service Credit Event Types](#) for more information.

If your [service credit membership type](#) indicates one or more subcategories to be associated with events for this type of membership, indicate the appropriate **Subcategory Name** for this event.

If your [service credit membership type](#) indicates that fiscal year is required, indicate the appropriate **Fiscal Year** for this event.

Enter the **Service Credit Event Amount**. If your [service credit membership type](#) indicates that your membership has a balance, this amount is used to calculate the overall balance for the membership as well as a balance for the subcategory and/or fiscal year, if applicable.

If this event was created as a result of a transfer from another membership, the **Transferred From Event ID** is displayed.

If the information about this event may be [downloaded to a third party](#) system, you should have an event completion algorithm that stamps the appropriate **Batch Control** and **Batch Number** onto the event.

The FT collection displays information about [financial transactions related to this event](#). Two different categories of financial transactions may be linked to the event.

- FTs that **contributed to** this event. These financial transactions played a part in the creation of this event. For example, if frequent flier miles are earned as a result of newly created bill segments, the FTs related to these bill segments are linked to the event and marked as **contributed to** FTs. This information is purely for audit purposes. It is the responsibility of the algorithm or background process that creates the event to populate the collection of **contributed to** FTs. Refer to [How Events Are Created](#) for more information.

**Note.** The amount of the **contributed to** FTs will not necessarily add up to the amount of the event. Refer to the algorithm or background process that calculates the amount of the event to understand how the event amount is calculated based on the amount of the **contributed to** FTs

- FTs that were **created by** this event. These financial transactions are created as a result of this event. For example, if you have generated a capital credit allocation event, perhaps you need to post information to the general ledger. An allocation event should therefore generate an adjustment to post the effect on the GL. The adjustment's FT is linked to the event as an audit. In order to create a financial transaction related to this event, you must plug in an appropriate event completion algorithm on your [service credit event type](#).

Refer to [An Event May Cause Other Actions to Occur](#) for more information.

# Appointments

**Separate module.** Please note that the Appointments functionality is associated with a separate [Appointments](#) 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. .

An appointment is an agreement with a customer to perform one or more field activities at a given premise on a specific date / time band. The topics in this section describe how you manage appointments and schedule appointments.

## Contents

- [The Big Picture of Appointments](#)
- [Appointment Maintenance](#)
- [Appointment Period Maintenance](#)

## The Big Picture of Appointments

---

The topics in this section provide background information about appointments.

## Contents

- [An Appointment Is Made With A Dispatch Group](#)
- [A Dispatch Group Has Appointment Periods](#)
- [An Appointment Is Notional](#)
- [Navigating To The Make / Break Appointment Page](#)
- [Appointments Are Not Applicable To All Field Activities](#)
- [Determining Available Appointments](#)
- [Limiting The Number Of Appointments In An Appointment Period](#)
- [Canceling An Appointment](#)
- [Compensating A Customer For A Missed Appointment](#)
- [Enabling Appointments](#)

## An Appointment Is Made With A Dispatch Group

A dispatch group is a logical group of work crews located at an operations area. When a [field activity is created](#), the system assigns it a dispatch group based on the type of activity, the type of service point, and the operations area that manages the service point. Refer to [Defining Field Order Options](#) and [The Big Picture Of Field Orders](#) for more information about field activity creation and dispatch groups.

When you make an appointment, you are booking time with one of the dispatch group's crews to perform one or more field activities located at a premise.

## A Dispatch Group Has Appointment Periods

An appointment period is a time-band on a given date range when an appointment can be scheduled with a dispatch group. For example, you might have an appointment period from 1 pm to 5 pm on 15-Jan-2003 for the "new service" crew.

A dispatch group can have many appointment periods on a given date. For example, if you offer appointments in two-hour time bands and your crews work 8 hours per day, then your dispatch groups will have four appointment periods (e.g., 8am to 10am, 10am to 12pm, 1pm to 3pm, and 3pm to 5pm).

The [Appointment Periods](#) transaction is used to maintain a dispatch group's appointment periods.

## An Appointment Is Notional

The [Appointment](#) transaction is used to make / break an appointment for a premise. When you make an appointment, you are actually linking one or more field activities to an appointment period. There is no entity in the system called "appointment". Rather an appointment is a group of field activities located at a premise that are linked to an appointment period.

## Navigating To The Make / Break Appointment Page

Any of the following methods can be used to navigate to the [Appointment](#) transaction:

- An alert in the [Alert Zone](#) highlights if an appointment exists for the current premise. The user is transferred to the appointment transaction if they drill down on the alert.
- You can setup an order's [campaign](#) to automatically transfer a user to the appointment transaction whenever an order is completed for a given campaign. This might be useful for campaigns that cause field activities to be created.
- A user can setup a [favorite link](#) associated with the appointment transaction.
- The [field activity](#) transaction allows a user to transfer to the appointment page when an appointment exists for a field activity
- Via any of the following menus:
  - Premise context menu
  - Field activity context menu
  - Main menu

## Appointments Are Not Applicable To All Field Activities

For each [FA type](#), you mark which types of field activities are eligible for dispatch. For those FA types that may be dispatched, you can indicate that an appointment is **required for dispatch**, **optional** or **not applicable**. Field activities for which appointments are not applicable are not displayed on the [appointment](#) transaction.

If your field activity requires an appointment but an appointment has not yet been made, the system does not allow you to associate the field activity with a field order.

## Determining Available Appointments

On the [Appointment](#) transaction, you can view available appointments for selected field activities by clicking a button. When this button is clicked, the available appointment periods are constructed by an algorithm plugged-in on the selected [dispatch group](#). This means your implementation can construct whatever rules it needs to determine the list of available appointments that are presented to a user. Refer to [AAVL-AVLAPPT](#) for a description of a base package plug-in.

## Limiting The Number Of Appointments In An Appointment Period

When you set up an appointment period for a dispatch group, you can define the maximum number of appointments that can be scheduled in the period. For example, you can indicate that the "meter installation crew for the northern region" has three appointments available between 1 pm and 5 pm on 15-Jan-2003.

**An appointment is notional.** As described above, an appointment is a group of field activities located at the same premise that are linked to an appointment period.

The appointment validation algorithm plugged-in on the [dispatch group](#) controls whether a new appointment can be scheduled at a premise. This means your implementation can construct whatever rules it needs to prevent overbooking an appointment period. Refer to [AAVL-AVLAPPT](#) for a description of a base package plug-in.

## Cancelling An Appointment

The [Appointment](#) transaction also allows a user to cancel an appointment. This simply breaks the linkage between a field activity and an appointment period.

The following points describe other mechanism that can cause appointment cancellation:

- If a user changes the start or stop date associated with a service agreement on the Start / Stop page, the system cancels all pending field activities and creates new ones. When the pending field activities are canceled, the appointments are implicitly canceled. The new field activities will NOT be linked to an appointment period when they are created (the user must make new appointments).
- If a user deletes a field activity, the appointment is implicitly canceled.

## Compensating A Customer For A Missed Appointment

In some locales, if an appointment is missed, the company has to pay a fine to the customer. No specific functionality exists in the system to support this practice. However, your implementation could implement this requirement as follows:

- Field activities that can have penalties should have a step that records a customer contact (note, a field activity's [field activity type](#) controls the steps that must be entered to complete a field activity).
- When a user adds such a contact, the contact date/time should be the date/time the crew showed up for the appointment.

- The field activity types associated with such field activities should have a completion algorithm that compares the appointment period's date/time to the customer contact's date/time and levies a penalty via the creation of an adjustment. Note, this algorithm is not supplied with the base package.
- In addition, you may want to create an implementation-specific table to record the linkage between the adjustment and the field activity.

## Enabling Appointments

This section describes set up tasks necessary to allow appointments to be made for a dispatch group.

### Contents

- [Plug In The Appropriate Algorithms On Each Dispatch Group](#)
- [Set Up Appointment Periods](#)

### Plug In The Appropriate Algorithms On Each Dispatch Group

Set up each dispatch group that can have appointments as follows:

- Define the appropriate "available appointments" plug-in on the dispatch group. Refer to [Determining Available Appointments](#) for more information.
- If you want to validate that an appointment period is valid for a field activity, plug-in the appropriate "appointment validation" algorithm on the dispatch group. Refer to [Limiting The Number Of Appointments](#) for more information.

### Set Up Appointment Periods

Use [Appointment Period Maintenance](#) to set up appointment periods for the dispatch group. Refer to [A Dispatch Group Has Appointment Periods](#) for more information.

## Appointment Maintenance

This page allows you to manage appointments for a premise's field activities. The following functions can be performed on this page:

- You can make an appointment for one or more field activities
- You can view existing appointments
- You can view historical appointments
- You can cancel an appointment

**The field activity may change.** When an appointment period is linked to a field activity, the field activity's scheduled date is changed to reflect the appointment date and the field activity's dispatch group is changed to reflect the appointment period's dispatch group.

Open this page using **Field Order, Appointment**.

### Description of Page

**Premise** is a concatenation of summary information about the premise. **Premise ID** is the unique identifier of the premise.

The **Existing Appointments** grid contains an entry for every field activity (regardless of status) that is linked to an appointment period at this **Premise**.

You can cancel an appointment for a field activity by clicking **Cancel**. The **Cancel** button is only enabled if the cancellation of the field activity would cause it to appear in the **Field Activities Without Appointments** grid (i.e., you can only cancel appointments for pending field activities).

**External System Integration.** If the [FA integration](#) module is not [turned off](#) and the dispatch group for this field activity references an [FA Integration](#) external system feature configuration, clicking **Cancel** triggers a real-time message to be sent to the external system to [cancel the appointment](#).

The **Field Activities Without Appointments** grid contains an entry for every field activity that meets the following qualifications:

- The field activity's status is [Pending](#).
- The field activity's activity type indicates that it is eligible for dispatch and that appointment booking is [required for dispatch](#) or [optional](#).
- The field activity isn't linked to a field order whose status is [Work Done, Not Recorded](#).
- The field activity isn't linked to an appointment period.

The **Search for an Appointment** section is used to view possible appointments for the selected field activities (and make an appointment by selecting of the available appointments).

**External System Integration.** The remaining information for this page is applicable to organizations that are not using the [FA integration](#) module, i.e. it is [turned off](#). If your organization uses this module, refer to [Appointments And External System Integration](#) for information about how this section may behave.

Click **Show Appointments** to populate the **Available Appointments** grid with appointment periods. You have the following options before you click this button:

- Specify a **Dispatch Group**. If you do this, available appointments for the specified dispatch group will be shown in the following grid.
- Leave **Dispatch Group** blank. If you do this, the system will look for available appointments for the “lowest common denominator” dispatch group shared by the selected field activities. Refer to [Using Alternate Dispatch Groups To Find The Lowest Common Denominator](#) for more information.
- Specify **On/Around Date**. If you do this, available appointments on / before this date will be shown in the following grid.
- Leave **On/Around Date** blank. If you do this, the system looks for available appointments on / before the earliest date from the selected field activities.

The **Available Appointments** grid is populated after you click **Show Appointments**. Note, an algorithm that's plugged-in on the [dispatch group](#) constructs the list of available appointments. Refer to [Determining Available Appointments](#) for more information.

**Assumptions at open.** The system selects all **Field Activities Without Appointments** and clicks **Show Appointments** when this page opens (assuming you want to schedule an appointment for these field activities). This can result in an error (e.g., if there is no common dispatch group amongst the selected field activities). When an error occurs, it is displayed adjacent to **Search for an Appointment**.

The **Available Appointments** grid contains an entry for every possible appointment period. In addition, adjacent to the grid is the **From** and **To** date range from which these appointment periods were extracted. You have the following options at this point:

- Click **OK** to create an appointment for the selected field activities. This simply links the selected field activities to the respective appointment period.
- Click the < or > buttons that surround the **From** and **To** dates. Clicking these buttons causes the system to look for appointment periods in the previous / next date range.

**The field activity may change.** When an appointment period is linked to a field activity, the field activity's scheduled date is changed to reflect the appointment date and the field activity's dispatch group is changed to reflect the appointment period's dispatch group.

**The field order may change.** When an appointment period is linked to a field activity that has been dispatched (i.e., linked to a field order), the field order's schedule date/time is updated with the earliest appointment date/time of all non-canceled non-completed field activities that are linked to the field order.

## Appointments And External System Integration

If the [FA integration](#) module is [turned off](#) and the [dispatch group](#) references an external system feature configuration, the **Search for an Appointment** section of the page is built based on the configuration of the external system [feature configuration](#) options.

For a **Dispatch Group** referencing an external system feature configuration:

The **From Date** defaults to the earliest scheduled date of the selected field activities. The **To Date** defaults to the From Date plus the value of the [\*\*Default Days of Available Appointment\*\*](#) feature configuration option whenever a dispatch group is selected.

If your external system allows [forced appointments](#), the **Forced** checkbox is displayed. The user should check this box when attempting to book an appointment that is not available in the system.

If your external system options include [user defined search criteria](#) fields, these fields are displayed below the from date and to date.

Click **Show Appointments** to invoke the external system to display available appointments. If the communication is successful, the **Available Appointments** grid displays the results.

Each available appointment includes an **OK** button to click to choose the desired appointment.

If your external system allows [forced appointments](#), or allows [manual appointments](#) is set to Y, then the grid includes "+" and "-" buttons to allow you to add or remove appointments.

If your external system allows [forced appointments](#), allows [manual appointments](#) or allows the user to [narrow the appointment window](#), then the start date / time and end date / time may be modified by the user.

If your external system options include [user defined result fields](#), additional columns are included after the end date / time for each user defined field.

When the user clicks **OK** for the desired appointment a message is sent to the external system to [book the appointment](#).

**The field activity may change.** If a message appointment is successfully sent, the field activity's scheduled date is changed to reflect the appointment date and the field activity's dispatch group is changed to reflect the appointment period's dispatch group.

Refer to [Booking Appointments Via An External System](#) for more information about this integration.

## Appointment Period Maintenance

An appointment period is a time-band on a given date range when an appointment can be scheduled with a dispatch group. For example, you might have an appointment period from 1 pm to 5 pm on 15-Jan-2003 for the "new service" crew. This page is used to maintain a dispatch group's appointment periods. Refer to [The Big Picture of Appointments](#) for more information about appointments and appointment periods.

**Mass update functionality exists.** This page has mass-update functionality that can be used to add / update / delete many appointment periods at once. Refer to [How to](#) for more information

**External System Integration.** If your organization uses the [FA integration](#) module, you do not need to set up appointment periods for dispatch groups that reference an external system. The system automatically creates [appointment period](#) records as an audit when an appointment is booked via the external system.

Open this page using **Field Order, Appointment Period**.

### Description of Page

The grid on this page contains one row for every appointment period linked to the **Dispatch Group**.

This grid is initially populated with appointment periods starting from today's date and ending 7 days in the future when you select a **Dispatch Group**. If you wish to see appointment periods for a different date range, enter a **Start Date** and/or **Number of Days** and then click the search button; the system will populate the grid with all appointment periods starting from **Start Date** and ending X days in the future (where X is **Number of Days**).

Each appointment period in the grid contains the following information:

- **Start Date** and **Start Time** define the start of the appointment period.

- **End Date** and **End Time** define the end of the appointment period. The end date is defaulted to the start date.
- **Maximum Appointments** defines the maximum number of appointments in the appointment period. Refer to [Limiting The Number Of Appointments](#) for more information about the uses of this field.
- **Scheduled Appointments** indicates the number of field activities that are currently linked to this appointment period. You can use the go to button to view the field activities in the appointment period.
- **Appt Period ID** indicates the unique identifier that has been assigned to this record.

Refer to [How To Copy Appointment Periods](#) for information about the **Copy Appointment Periods** button. This button is disabled if the dispatch group references a WFM system.

Refer to [How To Update Maximum Number Of Appointment Periods](#) for information about the **Update Max Appointments** button. This button is disabled if the dispatch group references a WFM system.

## How To

The topics in this section describe how to perform common appointment period functions.

### Contents

- [How To Copy Appointment Periods](#)
- [How To Update Maximum Number Of Appointments](#)
- [How To Delete Appointment Periods](#)
- [How To Overwrite Appointment Periods](#)

### How To Copy Appointment Periods

You can click **Copy Appointment Periods** to quickly create multiple appointment periods for a dispatch group by copying appointment periods from the same or a different dispatch group. The **Copy Appointment Periods** window opens when this button is clicked.

**Mass overwrites and deletes.** See the following description for how this pop-up can be used to overwrite and/or delete existing appointment periods.

In the **Copy Appointment Periods** window, you can:

Use **Copy From Dispatch Group** to define the dispatch group from which you want to copy appointment periods.

Use **Copy To Dispatch Group** to define the dispatch group for which you are creating appointment periods.

Use **Copy To Start Date** and **End Date** to define the period for which you are creating appointment periods. The system will display each date's day of the week and the **Total Days** in the period.

Use **Copy Method** to define how appointment periods will be copied from the **Copy From Dispatch Group** to the **Copy To Dispatch Group**. The following options are possible:

- Use ***Copy from the matching day in the template week*** if you want to copy the appointment periods from a matching day of the week from a template week. When this option is selected, you must identify the **Start Date of Template Week**. Adjacent to this date, the system will display:

- The day name of the start day
- The number of appointment periods in the start day
- The last day in the 7 day period (i.e., a day 7 days in the future)
- The day name of the last day
- The number of appointment periods in the last day

For example, if you enter a **Copy To** period of **1-Jan-2004** to **31-Jan-2004**, the system will create appointments for every Monday in this period by copying the appointment periods from the Monday in the **Template Week** (it will do the same for Tuesdays, Wednesdays, etc. in the **Copy To** period).

- Use ***Copy the same day repeatedly*** if you want to copy the appointment periods from a **Template Day**. Adjacent to this day, the system displays:

- The day name of the start day
- The number of appointment periods in the start day

For example, if you enter a **Copy To** period of **1-Jan-2004** to **31-Jan-2004**, the system will create appointments for every day in this period by copying the appointment periods from the **Template Day**.

Before you press **OK**, consider if holidays during the **Copy To** period should be treated differently than the template days. If so:

- Use **Calendar in which Holidays are Defined** to define the work calendar that the system should use to identify holidays. The system will display the number of holidays it finds in the **Copy To** period adjacent.
- Use **For Holidays, Copy the Periods from this Date** to define the template day whose appointment periods should be used for holidays.

After you have carefully verified this information, press **OK**. Please see the following notes if you detect a mistake after you press **OK**.

**You can also use this transaction to overwrite appointment periods.** This transaction is also capable of overwriting existing appointment periods for a dispatch group. If the **Copy To Dispatch Group** has appointment periods in the **Copy To** period, the system will ask if you want to delete them and then create fresh appointment periods. Please note, if there are any appointment periods in the **Copy To** periods with appointments, you may not overwrite the appointment periods.

**Mass Delete.** If the **Copy To** period has existing appointment periods (and they don't already have appointments) and the **Copy From Dispatch Group** doesn't have corresponding appointment periods, the **Copy To** periods will be deleted. Refer to [How To Delete Appointment Periods](#) for more information.

## How To Update Maximum Number Of Appointments

You can click **Update Max Appointments** to quickly update the maximum number of appointments on existing appointment periods. The window that appears is identical to that described under [How To Copy Appointment Periods](#); the only difference being that when this window is invoked in “update mode”, only existing periods linked to the **Copy To Dispatch Group** will be updated; appointment periods will never be created or deleted. Follow the following instructions when you need to update the maximum number of appointment periods.

Use **Copy From Dispatch Group** to define the dispatch group whose appointment period(s) contain the desired maximum number of appointments.

Use **Copy To Dispatch Group** to define the dispatch group whose appointment periods will be updated.

Use **Copy To Start Date** and **End Date** to define the period for which you are updating appointment periods. The system will display each date's day of the week and the **Total Days** in the period.

Use **Copy Method** to define how appointment periods will be updated. The following options are possible:

- Use **Copy from the matching day in the template week** if you want to update the appointment periods from a matching day of the week from a template week. When this option is selected, you must identify the **Start Date of Template Week**. Adjacent to this date, the system will display:
  - The day name of the start day
  - The number of appointment periods in the start day
  - The last day in the 7 day period (i.e., a day 7 days in the future)
  - The day name of the last day
  - The number of appointment periods in the last day

For example, if you enter a **Copy To** period of **1-Jan-2004** to **31-Jan-2004**, the system will update appointment periods for every Monday in this period by copying the maximum number of appointments from the Monday in the **Template Week** (it will do the same for Tuesdays, Wednesdays, etc. in the **Copy To** period).

- Use **Copy the same day repeatedly** if you want to update the appointment periods from a **Template Day**. Adjacent to this day, the system displays:
  - The day name of the start day
  - The number of appointment periods in the start day

For example, if you enter a **Copy To** period of **1-Jan-2004** to **31-Jan-2004**, the system will update appointments for every day in this period by copying the maximum number of appointment periods from the **Template Day**.

Before you press **OK**, consider if holidays during the **Copy To** period should be treated differently than the template days. If so:

- Use **Calendar in which Holidays are Defined** to define the [work calendar](#) that the system should use to identify holidays. The system will display the number of holidays it finds in the **Copy To** period adjacent.
- Use **For Holidays, Copy the Periods from this Date** to define the template day whose appointment periods should be used for holidays.

After you have carefully verified this information, press OK.

## How To Delete Appointment Periods

You can delete a dispatch group's appointment periods one-by-one by clicking the delete button in the appointment periods grid. To quickly delete many appointment periods for a dispatch group you can click **Copy Appointment Periods**. We understand this is counterintuitive. The following instructions describe how to do this:

Determine the period of time during which you want to delete appointment periods (note, you cannot delete an appointment period if there are any appointments linked to the appointment period). We refer to this period as the "deletion period" below.

Use **Copy From Dispatch Group** to define a dispatch group that doesn't contain appointment period(s) during the deletion period.

Use **Copy To Dispatch Group** to define the dispatch group whose appointment periods will be deleted.

Use **Copy To Start Date** and **End Date** to define the deletion period. The system will display each date's day of the week and the **Total Days** in the period.

Use the **Copy the same day repeatedly** Copy Method. You should identify a **Template Day** that has no appointment periods. Adjacent to this day, the system displays:

- The day name of the start day
- The number of appointment periods in the start day. Please confirm that this value is zero.

You don't have to specify the holiday period options.

After you have carefully verified this information, press OK.

## How To Overwrite Appointment Periods

The window that appears when you click **Copy Appointment Periods** has the ability to delete existing appointment periods (as long as they don't have appointments) and then create new ones. It works like this - if the **Copy To Dispatch Group** has appointment periods in the **Copy To** period, the system will ask if you want to delete them and then create fresh appointment periods. Please follow the instructions described under [How To Copy Appointment Periods](#) for a description of the **Copy Appointment Periods** functionality.

# Loans

There are many reasons why your company may loan money to customers. For example, your company may loan money to:

- Finance service upgrades (e.g., to change to dual fuel)
- Support economic development (e.g., to extend a power line)
- Promote energy conservation (e.g., customers borrow to insulate their house)

**Loans are optional.** You need only set up loans if your organization loans money to customers. For information on setting up the system to enable loans, refer to [Defining Loan Options](#).

## Contents

[The Big Picture Of Loans](#)

[Loan Maintenance](#)

[How To](#)

## The Big Picture Of Loans

---

The topics in this section provide background information about loans.

## Contents

[Loans Are Service Agreements](#)

[Loan Interest Charges](#)

[Loan Amortization Schedules](#)

[Loan Overpayments](#)

[The Lifecycle Of A Loan](#)

[Finalizing And Closing A Loan](#)

[Navigating To The Loan Maintenance Page](#)

## Loans Are Service Agreements

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, including the loan amount, periodic payment amount, the number of amortization periods, the interest rate, and the periodicity of the bills. Loan service agreements are just like other service agreements in many ways:

- The system bills the customer for the periodic payment amount using information on the service agreement.
- The customer pays the billed amount.
- If the customer doesn't pay, [credit and collections](#) may start a collection process to encourage the customer to pay (depending upon how the loan debt class has been set up on your system).

Loan service agreements differ from other service agreements in the following ways:

- Loans have [interest charges](#).
- Loans have [amortization schedules](#).

- Loan SAs are maintained on a separate transaction (refer to [Loan - Main](#) for more information).

**Loan service agreements are created using Start / Stop.** 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.

## Loan Interest Charges

When a bill is generated for a loan, the interest charge is calculated using an algorithm. The base package [loan interest charge algorithm type](#) calculates simple interest based on:

- Unbilled principal (i.e., the payoff balance minus the current balance)
- The number of billing periods covered by the bill
- The interest rate

**No interest on interest and no interest on past due amounts.** Just like a classic home loan, the base package algorithm does not calculate interest on interest, nor does it calculate interest on past due amounts. However, your system may be set up to levy a late payment charge.

After the interest charge is calculated, the bill segment FT algorithm generates the financial transactions for the bill. The base package FT algorithm generates a financial transaction that adds the interest charge to the payoff balance. In other words, if the customer wanted to pay off the loan, they would owe the principal balance plus the interest charge.

For more information, refer to [Billing For Loans And Interest Calculation](#).

## Loan Amortization Schedules

A loan's periodic payment amount is calculated such that the customer repays the loan in equal amounts over the term of the loan (with the exception of the first and last bill segments, which are typically for different amounts). 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).

The amortization schedule is a theoretical projection of the future. The term theoretical is used because the customer's actual payments can cause the schedule to change. For example, if a customer overpays a bill, the loan's length decreases and the amount of principal and interest in each scheduled payment changes.

A loan's amortization schedule is calculated when you click the **Calculate** button on [Loan - Main](#). When this button is clicked, an algorithm plugged in on the loan SA type actually calculates the amortization schedule.

You can view a loan's latest amortization schedule using [Loan - Main](#). You can also change the loan's payment periods or periodic payment amount on this page and recalculate the loan's amortization schedule.

### Negative Amortization

The system does not allow you to use a periodic payment amount that is less than the periodic interest amount. If it did, the customer's payoff balance would increase over time (this is referred to as "negative amortization"). There are two methods you can use to specify a payment amount in the system. Depending on which method you use, the system informs you of negative amortization differently:

- If you enter a payment amount that is too small on the Start confirmation page or in the **Renegotiate Future Payments** grid on [Loan maintenance](#), you get an error message when you click the **Calculate** button. You must fix the payment amount before you can start the loan or save your changes.
- If you manually override the **Payment Amount** on [Loan maintenance](#), you can save a payment amount that is too small to cover the interest charges. However, the system notifies you that it is not able to calculate the **Forecasted Schedule** because of negative amortization. Additionally, bill segments for the loan SA are produced in error until the payment amount is adjusted appropriately.

### 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. Typically, loan service agreements should be set up so that the overpayment is applied to the principal balance.

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 transfer the amount of the credit from the SA's current balance to its payoff balance, thus reducing the principal by the amount of the overpayment.

For more information about the adjustments created for loan overpayments, refer to [Overpayments On Loans](#).

**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.

### The Lifecycle Of A Loan

Loans have the same [lifecycle as other service agreements](#) (e.g., **pending start**, **active**, **pending stop**). You should rarely have to manually change a loan service agreement's status. Why? Just like other service agreements, the system changes the status behind-the-scenes when various events take place. For example, loans in the **pending start** state are automatically activated by a system background process.

If you need to manually change a loan service agreement's status (e.g., you need to cancel a loan that was inadvertently started), you can do so from [Service Agreement - Main](#).

## Finalizing And Closing A Loan

A loan SA is automatically stopped when the last bill segment is generated for the loan. If the remaining principal balance (plus accrued interest) is less than or equal to the loan payment amount, the SA status is changed to ***pending stop*** and the bill segment is marked as the closing bill segment. The SA can then be ***stopped*** by the user (manually using the SA page) or by the SA activation background process. When the customer pays the last bill, the loan SA is closed.

For more information about transitioning SAs from pending stopped, refer to [Finalizing Pending Stops](#).

## Navigating To The Loan Maintenance Page

Any of the following methods can be used to navigate to the [Loan Maintenance](#) transaction:

- A user can set up a [favorite link](#) associated with the loan transaction.
- Via any of the following menus:
  - Account context menu
  - Service agreement context menu
  - Main menu

**Alerts and Loans.** If the loan SA type is set up with an alert, the [Alert Zone](#) highlights if a loan exists for the current account. You are transferred to the SA maintenance transaction (not the loan maintenance transaction) if you drill down on the alert.

## Loan Maintenance

You can use loan maintenance to view the loan's amortization schedule and financial history as well as to change the terms of a loan, including the loan amount (prior to SA activation only), interest rate, payment amount and number of payment periods.

**Starting Loans.** You cannot start a loan from Loan - Main; you must use Start / Stop. Refer to [How To Start A Loan Service Agreement](#) for information.

Open **Customer Information, Loan** to maintain a loan.

### Description of Page

The fields at the top of the page are informational only. The **Payoff Balance** is the total amount owed by the customer (excluding any interest that may have accrued since the last bill). The **Current Balance** is the amount currently owed by the customer. The **Principal Balance** is the total principal amount left unbilled. Refer to [Payoff Balance and Current Balance for Loans](#) for more information.

The **Loan Amount** is the amount originally loaned to the customer. The **Standard Interest Rate** is the standard interest rate defined on the loan's SA type. If the SA type allows, the standard interest rate may be overridden by a start option and/or manually. The **Next Payment Amount** is the effective periodic payment amount as of the next bill.

The **Payment(s)** shows the number of payments that were used to calculate the original loan schedule.

The **Override Interest Rate** grid is used to override the standard interest rate. The interest rate can only be modified if the loan's SA type allows it. Note that changing the interest rate changes the loan amortization schedule. To override the interest rate:

- Specify an **End Date** for the override rate in effect if there is one.
- Click + and specify the **Start Date** and the new **Interest Rate**. You can also specify the **End Date** of the new override rate, but this is not required.

The **Payment Amount** grid is used to maintain the effective periodic payment amount. Note that changing the payment amount changes the amortization schedule. To change the payment amount:

- Click + and specify the **Effective Date** and the new payment **Amount**.

**Entering a payment amount that is too small.** If you save a payment amount that is too small to cover the periodic interest charges, this causes [negative amortization](#). The system is not able to recalculate the amortization schedule and bill segments are produced in error until the problem is corrected.

The **History** grid shows the loan's financial history in reverse chronological order. The financial history grid displays the following information:

- The **Arrears Date** is used to compute how many days old the debt is. In the context of a loan, debt only refers to the billed amount (the amount the customer needs to pay to stay current).
- **Financial Transaction Type** displays the type of financial transaction except for adjustments. For adjustments, the adjustment type's description is shown.
- **Current Amount** is the effect of the FT on the service agreement's current balance.
- **Current Balance** is the amount owed by the customer at the time of the transaction.
- **Payoff Amount** is the effect of the FT on the service agreement's payoff balance. The value is grayed out if it is the same as the current amount.
- **Payoff Balance** is the total amount owed by the customer. The value is grayed out if it is the same as the current balance.

**Warning!** If you do not understand the difference between payoff balance and current balance, refer to [Payoff Balance And Current Balance For Loans](#).

The **Forecasted Schedule** displays the loan amortization schedule. The first row of the schedule shows the current cumulative interest, current cumulative principal and the current principal balance. All other rows show the future amortization schedule based on the loan details. The forecasted schedule includes the following information:

- The **Loan Period** displays the forecasted payment period dates based on the billing frequency and the payment periodicity. The Loan Period is blank for the first row, which shows current cumulative amounts.
- The **Amount** displays the projected amount of that payment. The Amount is zero for the first row, which shows current cumulative amounts only.
- The **Interest** displays the projected amount of interest in the payment amount. The Amount is zero for the first row, which shows current cumulative amounts only.
- The **Principal** displays the projected amount of principal in the payment amount. The Amount is zero for the first row, which shows current cumulative amounts only.
- The **Cumulative Interest** is the total interest amount that would theoretically be billed at that loan period.
- The **Cumulative Principal** is the total principal amount that would theoretically be billed at that loan period.
- The **Principal Balance** is the remaining principal balance that would theoretically be remaining at the loan period.

**Forecasted schedule.** The loan schedule algorithm calculates the values in the forecasted schedule. Depending on the algorithm set up for your installation, the values may have a different meaning.

The **Renegotiate Future Payments** area allows you to change the terms of the loan. You can modify the following options:

- You can modify the **Loan Amount** only if the SA has not been activated. After the SA is activated, this field is for information only.
- The **Interest Rate** allows you to change the interest rate. When changing the interest rate, specify either the number of payment amounts or the payment amount and click Calculate to recalculate the amortization schedule.
- The **Number of Payment Periods** allows you to change the number of periods remaining in the loan's term. The **Frequency** displays the periodicity of the payments. If you change the number of payment periods, use the Calculate button to recalculate the payment amount.
- The **Payment Amount** allows you to change the amount of individual payments. If you change the payment amount, use the Calculate button to recalculate the number of periods.
- The **Calculate** button recalculates the payment amount (when you specify the interest rate and the number of payment periods) or the number of payment periods (when you specify the interest rate and the payment amount) and recomputes the loan amortization schedule based on these new terms.

**Implementing New Loan Terms.** The changes that you make using **Override Interest Rate**, **Payment Amount** and **Renegotiate Future Payments** do not take effect until you save the loan. Changes made to the **Renegotiate Future Payments** section are only valid for save after you click Calculate, and they override any other changes.

**Saving Renegotiated Future Payments.** When you save a loan that has renegotiated payments, the system may add new records for the override interest rate, payment amount, or both. It uses the start date of the next loan period as the start date of the new interest rate and/or as the effective date of the new payment amount. If the loan has not yet been billed, the system updates the interest rate and/or payment amount created when the loan was started.

## How To

The topics in this section describe how to perform common loan functions.

### Contents

- [How To Start A Loan Service Agreement](#)
- [How To Modify The Terms Of A Loan](#)
- [How To Adjust Loans](#)
- [How To Stop A Loan Service Agreement](#)

## How To Start A Loan Service Agreement

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.

Creating a loan service agreement involves the following steps:

- Select the account on Control Central then open Start/Stop.
- Enter a **Start Date** and a **Start Method** of **Start a SA**.
- Indicate the **CIS Division / SA Type** of the loan service agreement and click the **Start** button.
- After you click **Start**, the Start Confirmation page appears. When you select a loan service agreement type, the Start Confirmation page displays special fields that allow you to specify the loan's terms.
- Select a loan **Start Option**. The start option includes the adjustment type for booking the principal amount, and can also include defaults for the initial loan amount, periodic payment amount, payment periods, and interest rate. The system uses the information on the start option to populate the service agreement's default parameters. You can manually override these parameters:
  - Enter the **Loan Amount** (the amount that the customer is borrowing). Note that the label for the Total Amount To Bill (e.g., Loan Amount) may appear differently depending on how the SA type was defined.

- The system can automatically calculate either the payment amount (based on the loan amount and the payment periods) or the payment periods (based on the loan amount and the payment amount). Enter either the **Payment Amount** or the **Payment Periods** and click **Calculate**. Note that the label for the Recurring Charge (e.g., Payment Amount) may appear differently depending on how the SA type was defined.
- Once the loan parameters are populated, click **Start**.

The loan is created in the **pending start** state. The next time the SA activation process runs, it will activate the loan SA. If you want to activate the loan immediately, navigate to the service agreement page and click **Activate SA**.

**Overriding the Interest Rate.** The **Interest Rate** is a display only field. The interest rate is taken from the value of the interest bill factor defined on the SA type or the contract value of the interest bill factor on the start option if one is defined. After starting a loan, you can manually override the interest rate using [Loan - Main](#) if the SA type allows it.

**Changing The Loan Amount.** You can use loan maintenance to change the loan amount for an SA as long as the SA has not yet been activated. It may be helpful to transfer to [Loan - Main](#), and verify the amortization schedule before activating the SA.

## How To Modify The Terms Of A Loan

After a loan is active, you may want to modify its terms. You can change number of payment periods, the payment amount, and the interest rate (if allowed by the SA type). To modify the terms of a loan:

- Navigate to the [Loan maintenance](#) page.
- If you used the account context menu, the loan you want to modify may be displayed. If not, use the search dialog that appears to search for and select the loan you want to modify.
- In the **Renegotiate Future Payments** section, change the interest rate (if allowed), the number of payments, or the payment amount. Note that either the number of payments or payment amount must be blank.
- Click **Calculate**. The **Forecasted Schedule** is recalculated along with either the new payment amount or the new number of payments (depending on which was left blank).
- If the renegotiated payments are satisfactory to the customer, save the changes.

When you save the changes, appropriate entries are made in the **Override Interest Rate** and **Payment Amount** grids.

**Saving Loan Modifications.** You must save the renegotiated payments after clicking Calculate for the loan modifications to be effective.

## How To Adjust Loans

You issue ad hoc adjustments if you need to change a loan's principal and/or current balance outside of the normal billing and payment functions.

An adjustment can:

- Change a loan's payoff balance.
- Change a loan's current balance (i.e., how much the customer thinks they currently owe).
- Change both the loan's payoff and current balance.

For details about the adjustment types to apply, refer to [Adjusting Loan Amounts](#). For information about creating adjustments, refer to [Adjustments](#).

## How To Stop A Loan Service Agreement

Normally, a loan SA is automatically stopped when the bill segment for the last payment in the amortization schedule is generated for the loan (refer to [Finalizing And Closing A Loan](#)).

If a customer decides to payoff a loan before the term is complete, you can:

- Transfer to the SA maintenance page for the loan service agreement, specify an **End Date** for the loan, and click Save. This transitions the loan SA to ***pending stop***.

**End Date.** The loan SA cannot be stopped until the specified end date is reached, so do not specify an end date that is in the future unless you do not want to stop the loan until that date.

Once the loan is in pending stop and the end date has been reached, the system transitions the loan SA to ***stopped***. On the service agreement's normal bill cycle, the system final bills the SA for any outstanding balance plus accrued interest charges.

If you want to stop the loan and create the final bill immediately, complete these additional steps:

- On the loan's SA maintenance page, click the Stop SA button to transition the loan SA to ***stopped***.
- Manually, [create a bill](#) for the loan service agreement. Remember that if you only want to generate a bill segment for a specific service agreement (e.g., the loan service agreement), transfer to the bill segment page. From there, you can generate and freeze a bill segment for a specific service agreement. When finished, return to the bill page and complete the bill.

When the customer pays the final bill, the loan SA is closed by the system.

For more information about SA states, refer to [The Lifecycle Of A Service Agreement](#).

# Non-billed Budgets

A non-billed budget (NBB) is a payment plan used to level out a customer's payments over time. 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. For example:

- Rather than receive a bill once a quarter, a customer could set up an non-billed budget to pay fortnightly
- Rather than pay two large bills a year, the customer could set up an non-billed budget to pay 10 out of 12 months (sparing themselves during the holiday season)

Non-billed budgets can cover one or more service agreements linked to the same account. You can have multiple active non-billed budgets for an account, but any given service agreement can only be covered by a single non-billed budget at any point in time.

**Non-billed budgets are optional.** The business processes described in this section are only relevant if your organization offers non-billed budgets to its customers.

For information about setting up non-billed budget functionality for your system, refer to [Defining Non-billed Budget Options](#).

## Contents

- [The Big Picture Of Non-billed Budgets](#)
- [Maintaining Non-billed Budgets](#)
- [How To](#)

## The Big Picture Of Non-billed Budgets

The following topics provide background information about non-billed budgets.

## Contents

- [Non-billed Budget Service Agreements](#)
- [Which SAs Are Covered By A Non-billed Budget?](#)
- [Non-billed Budgets Versus Other Plans For Payment](#)
- [There Are Two Types Of Non-billed Budgets](#)
- [Navigating To The Non-billed Budget Maintenance Page](#)
- [Recommending Scheduled Payments](#)

## Non-billed Budget Service Agreements

Non-billed budgets are initiated by creating a non-billed budget service agreement for a customer. The non-billed budget SA and its SA type maintain the details of the non-billed budget, such as recommendation rules, covered SAs, scheduled payments and credit balance. Non-billed budgets are just like other service agreements in some ways:

- Non-billed budgets have a current and a payoff balance.

- If the customer doesn't pay a monitored non-billed budget, credit & collections may start a collection process to encourage the customer to pay (depending upon how the debt class has been set up on your system).

Non-billed budgets differ from other service agreements in the following ways:

- Non-billed budgets have a special role of non-billed budget.
- Non-billed budgets have scheduled payments.
- Non-billed budgets maintain a list of covered SAs (utility and other service agreements to which the non-billed budget applies)
- Non-billed budgets are started and maintained on a separate transaction (refer to [Non-billed Budget - Main](#) for more information).

## Which SAs Are Covered By A Non-billed Budget?

A non-billed budget can apply towards one or more of the account's billable SAs whose SA type is flagged as **Eligible for Non-billed Budget**. SAs can be added to or removed from a non-billed budget at any time. Additionally, an SA can be linked to more than one non-billed budget as long as the non-billed budgets to which it is linked do not have overlapping start and end dates.

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.

## Non-billed Budgets Versus Other Plans For Payment

Unlike a regular budget but like a payment plan, a non-billed budget has one or more scheduled payments. However, unlike payment plans, bills are not produced for the scheduled payment amounts. Instead, customers are simply notified of the payment schedule and it is their responsibility to make the scheduled payments. In general, the following statements highlight the differences between non-billed budgets and other payment plans.

- There is a payment schedule that is not dependent on billing, unlike [budget plans](#).
- Non-billed budgets are not limited to SAs of a single debt class, unlike [pay plans](#).
- Debt is not transferred from regular SAs to the non-billed budgets, unlike [payment arrangements](#).

**Non-billed Budgets and Budget Plans!** An SA should not be on a non-billed budget if it is on a [budget plan](#). A non-billed budget overrides any budget functionality in most cases. However, placing an SA that is on a budget on a non-billed budget may result in indeterminate results. For example, when the non-billed budget is broken, the stop algorithm for the non-billed budget syncs current and payoff balances for those utility SAs that were covered by the non-billed budget.

## There Are Two Types Of Non-billed Budgets

There are two types of non-billed budgets: monitored and unmonitored. As the terms suggest, the difference between monitored and unmonitored non-billed budgets is whether they are monitored by the account debt monitor. However, this also means that monitored and unmonitored non-billed budgets receive different financial treatment by the system.

In general, monitored non-billed budgets have their current balances monitored by the account debt monitor while the current balances of any covered SAs are zero (and therefore don't violate any debt collection criteria). With monitored non-billed budgets, customers make required prepayments that are applied towards their utility bills (until the non-billed budget is stopped). They are responsible for making sure the payments on their non-billed budget are up to date, but they do not have to payoff any balances on covered SAs.

Unmonitored non-billed budgets do not have their current balances monitored by the account debt monitor while the current balances of any covered SAs are maintained and therefore subject to the account debt monitor. When customers are set up on an unmonitored non-billed budget, the payments that they make towards the budget are optional. They are, however, financially responsible for paying the current balances of any covered SAs.

Refer to [Financial Transactions For Unmonitored Non-billed Budgets](#) for detailed information about the financial differences between monitored and unmonitored non-billed budgets.

**SA Type Determines Monitored Status.** The SA type that you select when setting up a non-billed budget determines whether it is monitored or unmonitored (i.e., it's a setting on the SA type). Your company may choose to implement only monitored non-billed budgets, only unmonitored non-billed budgets, or both.

## Navigating To The Non-billed Budget Maintenance Page

Any of the following methods can be used to navigate to the [Non-billed Budget Maintenance](#) transaction:

- An alert in the [Alert Zone](#) highlights if a non-billed budget exists for the current account. You are transferred to the non-billed budget transaction if you drill down on the alert.
- You can setup an order's [campaign](#) to automatically transfer a user to the non-billed budget transaction whenever an order is completed for a given SA. This might be useful for campaigns that are associated with services that are typically on a non-billed budget.
- A user can setup a [favorite link](#) associated with the non-billed budget transaction.
- Via any of the following menus:
  - Account context menu
  - Service agreement context menu
  - Main menu

## Recommending Scheduled Payments

When creating a non-billed budget, you must select a recommendation rule. The recommendation rule establishes the amount to be paid and the dates on which the payments are due. In other words, the recommendation rule creates a recommended payment schedule.

The recommendation rules that you can use are specific to your implementation. The recommendation rule that you select can have default values for its parameters. If your implementation allows, you can manually override one or more of these default values. For example, you may be able to specify the period covered by the non-billed budget and the day of the month on which the payments are due.

If your implementation allows, you can manually modify the recommended payment schedule to suit the needs of a particular customer.

Refer to [Designing Recommendation Rules](#) for more information.

## Maintaining Non-billed Budgets

Non-billed budget service agreements are started and maintained on a different transaction than other SAs. The topics in this section explain how to start and maintain non-billed budgets.

### Contents

- [Non-billed Budget - Main](#)
- [Non-billed Budget - History](#)

### Non-billed Budget - Main

The Main page contains basic information about the non-billed budget. Open **Customer Information, Non-billed Budget** to maintain this information.

#### Description of Page

**Non-billed Budget SA** (service agreement) information and **SA ID** are displayed on each page. These values only appear after the service agreement exists on the database. The **SA ID** is a system assigned random number that stays with a service agreement for life. The **Non-billed Budget SA** information is a concatenation of important details about the service agreement and its account.

The **Account ID** is identification of the account that is financially responsible for the service agreement.

Indicate the **CIS Division** and **SA Type**. These fields control aspects of the service agreement's behavior, such as whether the non-billed budget is monitored or unmonitored. These fields are gray when the status is not **Pending Start**.

For more information about non-billed budget SA types, refer to [Defining Non-billed Budget Options](#).

If **Scheduled Payment Auto Pay** is **Included in Auto Pay** and the account is set up for automatic payment, the scheduled payments are covered by auto pay. If **Excluded from Auto Pay**, the scheduled payments are excluded from auto pay if it is set up for the account.

**Recommendation Rule** is used to create a recommended payment amount and payment schedule. The SA type controls which recommendation rules are allowed for the non-billed budget SA.

**Current Balance** indicates the amount the customer currently owes on the non-billed budget. If the non-billed budget is unmonitored, the current balance is always zero.

**Payoff Balance** displays the total amount the customer would owe if they wanted to close the service agreement. For non-billed budgets, the payoff balance typically represents an accumulated credit that has not yet been distributed to the covered SAs (at bill completion time or when the non-billed budget is stopped).

The **Start Date** defines when the financial relationship begins. The **End Date** is a display only field that indicates when the financial relationship terminates. The **Expiration Date** is the date on which the SA expires. The **Renewal Date** is the date on which the SA is scheduled to be renewed by the SA renewal background process. If the SA is not renewed by the expiration date, the stop process is initiated.

The **Recommendation Rule Parameter Values** grid specifies the parameters and their values to use for the selected recommendation rule:

- **Parameter** is the name of the parameter from the recommendation rule algorithm type.
- **Value** is the value used for the parameter. If the recommendation rule allows, you can override the default values by specifying your own.

**Note.** The recommendation rule's [payment schedule algorithm type](#) controls the number and type of parameters. The recommendation rule controls the default values and whether they can be overridden.

The **Covered Service Agreements** grid lists the utility and other SAs that are on the non-billed budget. You must create the list of covered SAs before the recommendation rule can recommend a payment amount and list of scheduled payments. The grid displays the **SA ID**, **SA Information**, **Current Balance** and **Payoff Balance** for covered SAs.

The scheduled payments grid displays the scheduled payments created by the recommendation rule (after you click the Recommend NBB button). You can manually create or modify the **Scheduled Date** and or **Scheduled Amount** of any scheduled payment in the list. The **Total of Scheduled Payments** shows the total of all scheduled payments in the list.

Refer to [Scheduled And Actual Payments On The Non-billed Budget](#) for more information on what occurs when the base [non-billed budget scheduled payment background process](#) processes scheduled payments that are due.

In the Action grid:

- Click **Recommend** to create a recommended payment schedule based on the recommendation rule.
- Click **Renew** to manually renew the non-billed budget. This executes the SA renewal algorithm for the SA type and returns new renewal and expiration dates for the SA.
- Click **Break** to manually cancel the non-billed budget. This executes the break non-billed budget algorithm defined for the SA type, and changes the status of the SA to [\*\*pending stop\*\*](#).

## Non-billed Budget - History

The Financial History page displays the non-billed budget's historical financial transactions. Open **Customer Information, Non-billed Budget** and navigate to the **History** tab to view this information.

### Description of Page

**Non-billed Budget SA** (service agreement) information and **SA ID** are displayed on each page. These values only appear after the service agreement exists on the database. The SA ID is a system assigned random number that stays with a service agreement for life. The Non-billed Budget SA information is a concatenation of important details about the service agreement and its account.

The **Account ID** identifies who is financially responsible for the service agreement.

The grid shows the non-billed budget's financial history in reverse chronological order, including:

- The **Arrears Date** is used to compute how many days old the debt is. For monitored non-billed budgets, debt refers to the scheduled payment amounts (i.e., the customer only owes the scheduled payment amounts up to the most recent scheduled payment date). The arrears date for a schedule payment is the scheduled payment date.
- **Financial Transaction Type** displays the type of financial transaction except for adjustments. For adjustments, the adjustment type's description is shown.
- **Current Amount** is the effect of the FT on the service agreement's current balance.
- **Current Balance** is the amount owed by the customer at the time of the transaction.

## How To

---

The topics in this section describe how to perform common non-billed budget functions.

### Contents

- [How To Start A Non-billed Budget Service Agreement](#)
- [How To Cancel / Stop Non-billed Budget SAs](#)
- [How To Renew Non-billed Budgets](#)
- [How To Set Up Automatic Payment For A Non-billed Budget](#)

## How To Start A Non-billed Budget Service Agreement

To set up a new non-billed budget:

- Navigate to the Non-billed Budget maintenance page.
- If the account is not already specified, select the **Account** for which you want to start the non-billed budget.
- Define the **CIS Division** and **SA Type** to use for the non-billed budget service agreement.
- If necessary, change the **Start Date** (the current date is defaulted).
- Select the **Recommendation Rule** to use and modify any of the default parameters to meet the needs of the customer.

- Verify the list of **Covered Service Agreements**. All the account's SAs that are eligible for non-billed budget are listed. Click  to remove unwanted SAs from the list.
- Click **Recommend**.
- On the pop-up dialog, specify the **Schedule Start Date** and click **Recommend**.
- Make any necessary changes to the expiration date, renewal date and scheduled payments grid and click **Save**.

## How To Cancel / Stop Non-billed Budget SAs

To stop a non-billed budget:

- Navigate to the Non-billed Budget maintenance page.
- If the account with the non-billed budget you want to stop is not specified, select the **Account** of the non-billed budget you want to stop.
- If the account has more than one non-billed budget SA, select the non-billed budget you want to stop.
- Click **Break**.

## How To Renew Non-billed Budgets

Most non-billed budgets are renewed automatically by a background process. However, you may manually renew non-billed budgets that are eligible for renewal if desired.

To renew a non-billed budget:

- Navigate to the Non-billed Budget maintenance page.
- If the account with the non-billed budget you want to renew, select the **Account** of the non-billed budget you want to stop.
- If the account has more than one non-billed budget SA, select the non-billed budget you want to renew.
- If desired, change the recommendation rule or any of the parameters.
- Click **Renew**.

The new expiration and renewal dates are calculated and the new payment schedule is appended to the list of scheduled payments.

## How To Set Up Automatic Payment For A Non-billed Budget

If a customer wants to pay their non-billed budget scheduled payments automatically:

- Set up the account for automatic payment (as described under [How To Set Up A Customer To Pay Automatically](#)).
- Navigate to the Non-billed Budget Maintenance page and select the non-billed budget for the account.

- Set the **Scheduled Payment Auto Pay** field to indicate that the scheduled payments are *Included in Auto Pay*.

# Asset Inventory

In addition to maintaining a simple list of your meters and items, you may want to keep track of their locations when they are not installed at a service point. The topics in this section describe how you set up stock locations and manage your asset inventory.

**Asset.** We use the term “asset” to refer to both meters and items in the following discussion.

**Equipment is not tracked.** The functionality provided for tracking items that are installed at a service point are specifically referring to items that are linked to the service point via [SP / item installation](#). Items that are linked to service points as equipment are not included in this functionality.

## Contents

- [The Big Picture Of Asset Inventory](#)
- [Setting Up Asset Inventory Options](#)
- [Maintaining Stock Locations](#)
- [Maintaining Stock Location Inventory](#)

## The Big Picture Of Asset Inventory

The topics in this section provide background information about keeping track of the stock locations at which meters and items are located when they aren't installed at a service point.

## Contents

- [The Big Picture Of Stock Locations](#)
- [Stock Location History Is Created Behind The Scenes](#)
- [A User May Manually Change An Asset's Stock Location](#)
- [Viewing What Is At A Stock Location](#)
- [Viewing Where An Asset Is / Was](#)
- [How To Handle A Shipment Of New Assets](#)
- [How To Transfer Assets To Another Location](#)

## The Big Picture Of Stock Locations

There are two types of stock locations:

**Inventory stock location.** An inventory stock location is one at which an asset can reside when it's not installed at a service point.

**Parent location.** Stock locations that are used to organize inventory locations are called “parent” stock locations. You would create parent locations to provide a hierarchical perspective as to the location of your assets.

For example, assume you have a large warehouse with many different logical areas in which assets can be stored. You could create an inventory stock location for each specific shelf / storage area in the warehouse and then group all of these inventory locations under a parent location (where the parent location would be the “large warehouse”).

**Parent locations are not necessary.** Parent locations are only needed if you want to view stock locations in a hierarchy on the [Stock Location Inventory](#) page.

## Stock Location History Is Created Behind The Scenes

An asset's stock location history is only maintained if the asset's respective [meter type](#) / [item type](#) indicates that stock location history is maintained. If stock location is tracked, the system will create stock location history records in the following situations:

- Whenever an asset is installed at a service point, a stock location history record is stored. The stock location is blank on such a record (because the asset is installed at a service point).
- Whenever an asset is removed from a service point, an "installation" stock location history record is stored. The default stock location into which a removed asset is placed is defined on [Postal Code Default - Service Defaults](#).
- Whenever a meter or item is replicated, the operator must indicate the stock location at which the replicated meters / items reside. Refer to [Meter Replicator](#) and [Item Replicator](#) for more information.

**Note.** If location tracking for an item type or meter type is turned off and then subsequently turned on again, any change in the location of the item / meter during the period in which location tracking is off will not be recorded. This will result in gaps in the asset's location history.

## A User May Manually Change An Asset's Stock Location

[Stock Location History Is Created Behind The Scenes](#) most of the time. However, the following points describe situations that will cause an operator to manually change an asset's stock location.

- If an asset is not installed at a service point, an operator may change its current stock location using either [Meter - Location History](#) or [Item - Location History](#). This would typically be done when an asset is transferred to a different stock location.
- Whenever a new meter / item is initially added, the operator must indicate the stock location at which the meter / item resides. This is done using either [Meter - Location History](#) or [Item - Location History](#).

## Viewing What Is At A Stock Location

You can view all assets at a stock location using [Stock Location Inventory](#).

## Viewing Where An Asset Is / Was

You can use [Meter - Location History](#) and [Item - Location History](#) to view the history of the service points and stock locations at which an asset has been located. It's important to be aware that you may only view this history if the asset's [meter type](#) / [item type](#) indicates that the inventory location is tracked.

## How To Handle A Shipment Of New Assets

If you receive a shipment of new items, follow the following steps:

- Create a new [item](#) for each type of item that's received. These items serve as the "templates".
- Go to the [Item Replicator](#) to create multiple copies of the template item(s). Before you are allowed to replicate the new item, you must define the stock location at which the items will be stored.

If you receive a shipment of new meters, follow the following steps:

- Create a new [meter](#) for each type of item that's received. These meters serve as the "templates".
- Add a meter configuration for each template meter using [Meter Configuration Maintenance](#).
- Go to the [Meter Replicator](#) to create multiple copies of the template meter(s). Before you are allowed to replicate the new meter, you must define the stock location at which the meters will be stored.

## How To Transfer Assets To Another Location

To move assets from one stock location to another, perform the following steps:

- Navigate to either [Stock Location Inventory - Item Inventory](#) or the [Stock Location Inventory - Meter Inventory](#) as appropriate and select the stock location where the assets that you want to transfer are located.
- Display the items / meters that you want to transfer using the filters and pressing the search button.
- Select the items / meters that you want to transfer.
- Click on the **Transfer Location** button. The **Transfer Stock Location** window opens.
- Specify the **Stock Location** where you want to transfer the assets to and enter the effective date/time of the transfer in **Location Date/Time**. Click on the **Transfer** button to perform the transfer.

## Setting Up Asset Inventory Options

---

This section describes tables that must be set up before you can associate meters / items with inventory locations.

### Contents

- [Setting Up Postal Defaults](#)
- [Turn On Stock Location Tracking On Your Meter Types / Item Types](#)

## Setting Up Postal Defaults

On [Postal Code Default](#), define the respective stock location into which a meter / item will be placed by default when its removed from a service point. Refer to [Stock Location History Is Created Behind The Scenes](#) for more information.

## Turn On Stock Location Tracking On Your Meter Types / Item Types

Enable stock location tracking for the [meter types](#) / [item types](#) whose stock location is to be tracked in the system.

## Maintaining Stock Locations

Every meter / item whose [meter type](#) / [item type](#) indicates that its stock location is tracked resides at a stock location. The topics in this section describe how to maintain your stock locations.

### Contents

- [Stock Location - Main](#)
- [Stock Location - Characteristics](#)

### Stock Location - Main

Open this page using **Meter, Stock Location** and then navigate to the **Main** tab.

#### Description of Page

A stock location is a physical or logical location at which meters / items are stored while they are not installed at a service point. Refer to [The Big Picture Of Stock Locations](#) for more information.

Enter a unique ID and a **Description** for each **Stock Location**.

Indicate that this stock location is **Allowed** to **Hold Inventory** if it is one at which meters and items can be stored. Otherwise, indicate that it is **Not Allowed**.

**Note.** **Hold Inventory** is typically **Not Allowed** on “notional” stock locations used to group together detailed stock locations. You can view a stock location’s subsidiary locations in the tree at the bottom of this page.

If the **Stock Location** is subsidiary to a higher-level stock location, define its immediate **Parent Stock Location**. Refer to [The Big Picture Of Stock Locations](#) for more information about parent locations.

Enter any **Comments** about the stock location.

The [tree](#) at the bottom of the page shows the hierarchy of stock locations associated with the **Stock Location** selected in the search.

#### Where Used

Stock location history records reference a stock location. Refer to [Stock Location History Is Created Behind The Scenes](#) for more information.

### Stock Location - Characteristics

The characteristics page contains information that describes miscellaneous information about the stock location. Open this page using **Meter, Stock Location** and then navigate to the **Characteristics** tab.

### Description of Page

Use the **Characteristics** collection to define miscellaneous information to be kept for stock locations. For example, a premise ID can be specified as a characteristic to identify the address of a stock location.

**Note.** You can only choose characteristic types defined as permissible on the stock location record. Refer to [Setting Up Characteristic Types & Their Values](#) for more information.

To modify a characteristic, simply move to a field and change its value. To add a new characteristic, click + to insert a row, then fill in the information for each field. The following fields display:

<b>Effective Date</b>	Define the date on which the characteristic becomes effective. The effective date defaults from the <a href="#">Installation Record</a> .
<b>Characteristic Type</b>	Indicate the type of characteristic.
<b>Characteristic Value</b>	Indicate the value of the characteristic.

## Maintaining Stock Location Inventory

The topics in this section describe how to view and maintain the contents of a given stock location.

Refer to [The Big Picture Of Asset Inventory](#) for more information.

### Contents

- [Stock Location Inventory - Main](#)
- [Stock Location Inventory - Item Inventory](#)
- [Stock Location Inventory - Meter Inventory](#)

## Stock Location Inventory - Main

This page contains a tree that shows:

- A summary of the number of items and meters located at a stock location (that can hold inventory).

Refer to [The Big Picture Of Asset Inventory](#) for more information.

Open this page using **Meter, Stock Location Inventory** and then navigate to the **Main** tab.

### Description of Page

This page contains a [tree](#) that shows a summary of the types of meters and items stored at the location. Refer to [The Big Picture Of Stock Locations](#) for more information.

If you click an **Item Count** node, you will be transferred to the [Stock Location Inventory - Item Inventory](#) page where you can see the details of the individual items of a given type located at the **Stock Location**.

If you click on a **Meter Count** node, you will be transferred to the [Stock Location Inventory - Meter Inventory](#) page where you can see the details of the individual meters of a given type located at the **Stock Location**.

## Stock Location Inventory - Item Inventory

This page is used to display the individual items of a given type that are stored at a stock location. Refer to [The Big Picture Of Asset Inventory](#) for more information.

Open this page using **Meter, Stock Location Inventory** and then navigate to the **Item Inventory** tab.

### Description of Page

This page displays the individual items that are stored at a **Stock Location** as at a given **Date/Time**.

The easiest way to display inventory for a given **Item Type** at a **Stock Location** is to navigate to the [Stock Location Inventory - Main](#) tab and find the **Stock Location**. Then, click on an **Item Count** line.

You can restrict the list using the following filter options:

- Use the **Item Type** filter to restrict the meters to those of the given meter type.
- Use the **Badge Number** filter to restrict the list to meters that have a badge number beginning with or matching this value.

Don't forget to click the search button after changing the filters.

The **Select All / Clear All** buttons are used to select items to be transferred to another stock location. These buttons are enabled if at least one item is displayed in the grid.

The grid contains the item inventory at the selected stock location on the given date/time that match your filter criteria. The following information is displayed in the grid:

- Select box. Use this checkbox to select items that you want to transfer to another stock location. This checkbox is enabled if the item is currently stored at this location and location is tracked for the item's item type. Refer to [How to Transfer Assets To Another Location](#) for more information.
- The description of the item's **Item Type**.
- The item's **Badge Number**.
- The item's **Serial Number**.
- The **Manufacturer/Model** of the item.
- The **Location Date/Time** when the item was added to this stock location.
- If the item was subsequently removed from this stock location, the **Subsequent Location** column shows either the stock location it was transferred to or the service point it was installed at.

Pressing the **Transfer Location** button enables you to transfer the selected items to another stock location. This button is enabled if at least one meter in the grid is selected. Refer to [How to Transfer Assets To Another Location](#).

## Stock Location Inventory - Meter Inventory

This page is used to display the individual meters of a given type that are stored at a stock location. Refer to [The Big Picture Of Asset Inventory](#) for more information.

Open this page using **Meter, Stock Location Inventory** and then navigate to the **Meter Inventory** tab.

### Description of Page

This page displays the individual meters that are stored at a **Stock Location** as at a given **Date/Time**.

The easiest way to display inventory for a given **Meter Type** at a **Stock Location** is to navigate to the [Stock Location Inventory - Main](#) tab and find the **Stock Location**. Then, click on a **Meter Count** line.

You can restrict the list using the following filter options:

- Use the **Meter Type** filter to restrict the meters to those of the given meter type.
- Use the **Badge Number** filter to restrict the list to meters that have a badge number beginning with or matching this value.

Don't forget to click the search button after changing the filters.

The **Select All / Clear All** buttons are used to select meters to be transferred to another stock location. These buttons are enabled if at least one meter is displayed in the grid.

The grid contains the meter inventory at the selected stock location on the given date/time that match your filter criteria. The following information is displayed in the grid:

- Select box. Use this checkbox to select meters that you want to transfer to another stock location. This checkbox is enabled if the meter is currently stored at this location and location is tracked for the meter's meter type. Refer to [How to Transfer Assets To Another Location](#) for more information.
- The description of the meter's **Meter Type**.
- The meter's **Badge Number**.
- The meter's **Serial Number**.
- The **Manufacturer/Model** of the meter.
- The **Location Date/Time** when the meter was added to this stock location.
- If the meter was subsequently removed from this stock location, the **Subsequent Location** column shows either the stock location it was transferred to or the service point it was installed at.

Pressing the **Transfer Location** button enables you to transfer the selected meters to another stock location. This button is enabled if at least one meter in the grid is selected. Refer to [How to Transfer Assets To Another Location](#) for more information.

# Case Management

**Separate module.** Please note that the Case Management functionality is associated with a separate **Case 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.

Your organization can use case management functionality to manage a variety of situations, such as:

- a high-bill complaint,
- a bankruptcy,
- an inspection of a premise,
- a customer's request for literature,
- a contractor's request to extend a line,
- a customer's rejection of a quote,
- a customer's request to change information on a future date,
- the processing of a market message in a deregulated environment

**Your implementation controls how your cases behave.** Since virtually all aspects of case management functionality are controlled by your implementation, you can use it to handle a myriad of business requirements. Refer to [Defining Case Management Options](#) for the details.

## Contents

[Background Topics](#)  
[Maintaining Cases](#)

## Background Topics

The topics in this section provide background information about case management functionality.

## Contents

[Case Type](#)  
[Creating Cases](#)  
[Log Information](#)  
[Alerts](#)  
[To Do's and Cases](#)  
[Reports](#)

## Case Type

A case is controlled by the case type that it references. Some aspects of the case that are controlled by the case type include:

- The [lifecycle](#) of the case

- The [applicability](#) of a person, account, and/or premise on the case
- Which [fields are required](#) during the various stages of a case's life
- Which users can [access cases](#) of a given type
- Additional processing that should take place when a case enters or leaves a given state

Refer to [Case Type Controls Everything](#) for a description of the business rules that you control when you set up your case types.

## Creating Cases

To create a case, open the [case management](#) page in add mode, select the appropriate case type, fill in the other necessary fields, and click save. Oracle Utilities Customer Care and Billing also includes scripts that can help create cases. Refer to [Scripts and Cases](#) for more information.

You can also develop plug-ins and user exits to so that the system will create cases when specific events take place. For example, you could set the system to create a case when an umbrella agreement is ready for renewal or when a quote is created.

Please refer to [Maintaining Cases](#) for a detailed description of the case page and field descriptions.

## Log Information

A case log contains an entry for every recorded event during the lifecycle of the case. There are two types of log entries:

- **Automatic entries.** The system automatically creates an entry in the log when a case is created or there is a [status change](#). Users cannot modify or delete these log entries.
- **Manual entries.** Users can add manual entries to record significant events at their discretion.

Refer to [Case - Main](#) for the page used to view the log.

Refer to [Case Statistics for a Given Status](#) for information about a report that uses the log to provide productivity statistics.

## Alerts

The [alert zone](#) will highlight if the person / account / premise in context have cases in a state that is configured as "alertable" if you plug-in the appropriate algorithm on the installation record.

Refer to [Alert Info Is Controlled By An Installation Plug-In](#) for the details.

## To Do's and Cases

You can configure the system to inform users of cases that require their attention. Refer to [To Do's and Cases](#) for the details.

## Reports

Several reports summarize case information. Refer to the [Reports](#) chapter for the details.

# Maintaining Cases

Your organization can use the case management functionality to manage a variety of situations, such as:

- a high-bill complaint,
- a bankruptcy,
- an inspection of a premise,
- a customer's request for literature,
- a contractor's request to extend a line,
- a customer's rejection of a quote,
- a customer's request to change information on a future date,
- the processing of a market message in a deregulated environment

Since an implementation can configure virtually every aspect of a case, the look and feel of the case management page are dynamic based on how [case types](#) have been configured.

## Contents

- [Case - Main](#)
- [Case - Case Portal](#)
- [Case - Log](#)

## Case - Main

Open the case management page by selecting **Customer Information**, **Case** from the main menu.

### Main Information

**Case Info** contains a concatenation of important information about the case. **Case ID** is the system-assigned unique identifier of the case. These values only appear after the case is added to the database.

**Formatting may be performed by a plug-in.** Refer to [Case Info May Be Formatted By An Algorithm](#) for more information.

**Case Type** defines the type of case. This field is protected after the case has been added. Refer to [Case Type Controls Everything](#) for more information.

**Status** shows the current state of the case. Refer to [Case Lifecycle](#) for more information.

If the case has any To Do entries associated with it, a hyperlink will appear summarizing the number and status of the entries. You can select the link to view these entries. Refer to [To Do's and Cases](#) for more information.

**Date / Time Opened** shows when the case was created. If the case is in a final state, the **Date / Time Closed** will also be shown. Refer to [One Initial State and Multiple Final States](#) for more information.

**Script** references the business process assistant script that can help you work on the case. You can click on the hyperlink to execute this script. This field is hidden if no script is associated with this case's current Status. Refer to [Scripts and Cases](#) for more information.

The **Actions** section contains buttons you can use to action the case. This area is suppressed if no actions can be performed for the case given its current status. A button will be gray if you do not have security rights to the action. Refer to [Buttons Are Used To Transition A Case](#) for more information.

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. If the case has been automatically transitioned to its current state the following message area appears right after the action area. If an error is encountered in the process a corresponding error message is displayed in the Auto-transition Message Area. Refer to [Automatic Transition Rules](#) for more information.

**Note.** The Auto-transition Message Area only appears when a case has just been automatically transitioned to its current state online. It only remains visible until the data for the case is refreshed.

Use the **Comments** area to describe anything interesting or unusual about the case.

**Person** references the person associated with this case. This field is suppressed if the Case Type indicates that a person is not allowed on cases of this type. This field is protected if the case is closed (i.e., if its status is defined as being final). This value is defaulted to the Person currently displayed in the Dashboard.

**Account** references the account associated with this case. This field is suppressed if the Case Type indicates that an account is not allowed on cases of this type. This field is protected if the case is closed (i.e., if its status is defined as being final). This value is defaulted to the Account currently displayed in the Dashboard.

**Premise** references the premise associated with this case. This field is suppressed if the Case Type indicates that a premise is not allowed on cases of this type. This field is protected if the case is closed (i.e., if its status is defined as being final). This value is defaulted to the Premise currently displayed in the Dashboard.

**Responsible User** references the user who has overall responsibility for the case. This field is suppressed if the Case Type indicates that a responsible user is not allowed on cases of this type. This field is protected if the case is closed (i.e., if its status is defined as being final).

The **Contact Information** area is used to define how to contact the individual who initiated the case. These fields are suppressed if contact information is not allowed on cases of this type. The fields cannot be modified if the case is in a closed status. The following information can be defined.

- **Contact Person** references the person who should be contacted. This defaults to the Person defined above.
- **Preferred Contact Method** indicates how the person prefers to be contacted. The following options are available:
  - **Email.** If this method is selected, the Contact Person's email address appears adjacent.
  - **Fax.** If this method is selected, the Contact Person's fax number appears (the system knows which of a [person's phone numbers](#) is a fax number by the phone type).

- **Not Applicable.** If the person doesn't want to be contacted (or the contact method is unusual), use this option. You might want to consider describing the situation in the **Contact Instructions**.
- **Phone.** If this method is selected, the Contact Person's phone number appears.
- **Postal.** If this method is selected, the Contact Person's address appears. Refer to [Where Does The Address Come From](#) for a description of which address the system displays.
- **SMS.** If this method is selected, the Contact Person's short message service (SMS) information displays.

**Contact Instructions** indicates any special instructions indicated for when or how to return a call.

**Callback Phone Type**, **Callback Phone Number**, and **Extension** indicate the actual phone number and phone type that should be used to call.

### Characteristics

The **Characteristics** grid contains fields that contain important information about the case. This grid is suppressed if the Case Type does not use characteristics. This grid is protected if the case is closed (i.e., if its status is defined as being final).

You can only choose **Characteristic Types** defined as permissible for this case type. Refer to [Additional Information](#) for more information. If you attempt to use an **Action** to transition a case into a status that requires additional information, a pop-up will prompt you for the additional fields. Refer to [Required Fields Before A Case Enters A State](#) for more information.

## Case - Case Portal

To view additional information associated with a case, open by selecting **Customer Information**, **Case** from the main menu and then the **Case Portal** tab.

### General Information

**Case ID** is the system-assigned unique identifier of the case.

**Case Info** contains a concatenation of important information about the case.

### Object Portal

When information associated with the case has been stored in the case Character Large Object (**CLOB**) field (such as the data associated with a market message), the **Object Portal** displays this information as-is. You cannot modify the case CLOB field data from this page, but case type algorithms can be configured to work with this data. For information on mapping data into CLOB fields, refer to the Business Object tips document **Schema Nodes and Attributes**.

## Case - Log

To view the log entries associated with a case, open by selecting **Customer Information**, **Case** from the main menu and then the **Log** tab.

### Description of Page

The **Log** grid displays log entries (in reverse chronological order) that audit the progress of the case. Please note the following about the entries in this grid:

- The system automatically creates log entries when a case is created, every time its status changes, or whenever an error occurs during the transition of a case. Algorithms plugged into a case type state can also be configured to create log entries. You cannot modify or delete the log entries.
- You can manually add a log entry by pressing the + button and enter the **Details**. You cannot modify or delete this information after saving it.
- The following information appears in the grid:
  - **Log Date/Time** contains the date and time the log entry was created.
  - **Details** contains user-specified or system-generated information about the log entry.
  - **Related Object** is populated on log entries that were created to record the creation of some other object. For example, if a case plug-in creates a customer contact, the related object contains information about the customer contact. Please note that if the object description is shown in blue, you can click on the object's description to drill down to the object.
  - **Log User** contains the user who caused the log entry to be created.
  - **Log Type** indicates how the log entry was created. The possible values are:
    - **Created**. Only one entry per case will be of this type. This entry is automatically created by the system when the case is created.
    - **User Details**. A user creates this type of entry manually.
    - **Status Transition**. This type of entry is created by the system each time the case's status is changed (as a result of pressing one of the Action buttons).
    - **System**. An algorithm on the case type state may be designed/configured to create this type of entry when it does something auditable. For example, an algorithm that creates a customer contact could insert a log entry to show that it did this (it will also populate the FK to the customer contact, which will appear in the **Related Object** column).
    - **Transition Error**. The background status transition process creates an entry of this type if an error occurs when automatically updating a case to a subsequent status (as directed by either an Enter Status algorithm or an Auto-Transition algorithm). An entry of this type lists the case states involved with the failed transition, and the algorithm in which the error was generated.
    - **Exception**. The background status transition process creates an entry of this type under the same circumstances as a Transition Error. A log entry of this type indicates why the background transition failed.

**Note.** The values for this field are customizable using the Lookup table. This field name is CASE\_LOG\_TYPE\_FLG.

# Premise Management

**Separate module.** Please note that the Premise Management functionality is associated with a separate **Premise 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 premise management functionality facilitates the following:

- Grouping premises together under a single parent premise
- Quick & efficient setup and maintenance of apartment complexes and high-rises
- A single view of properties for a landlord, including service status
- Ability to change the landlord or service status for a group of premises in one transaction
- And more...

The topics in this section describe premise management functionality.

## Contents

[Overview](#)

[Premise Management Page](#)

## Overview

---

The topics in this section provide background information about premise management functionality.

## Contents

[Define Premise Hierarchy](#)

[Manage Premises for a Landlord](#)

[Manage Groups of Premises](#)

## Define Premise Hierarchy

Premise management allows you to define premise hierarchy. For example, you could also have the notion of

- A building
  - Multiple floors
    - Multiple apartments per floor

To define this hierarchy you must create a [premise](#) for each level in the hierarchy and for each "child" premise, indicate its appropriate parent premise. Once you have set up your premise hierarchies, you may view the hierarchies on trees on the premise page and on the control central [premise tree](#).

In addition, the system provides the following [alerts](#):

- When viewing a parent premise, an alert indicates how many child premises exist

- When viewing a child premise, an alert indicates that it is linked to a parent premise.

## Manage Premises for a Landlord

The premise management page provides the ability to view all the premises for the landlord along with information about the accounts linked to each premise.

You may use this page to assign, change or remove the landlord agreement for multiple premises in a single action. For example if a landlord sells his properties to another person, this page facilitates updating all premises with one action rather than changing the landlord for each premise separately. For example, if the landlord sells only a portion of his properties.

This page may also be used to start or stop service for one or more premises for the landlord. For example, a landlord plans to sell a high-rise apartment complex with 100 apartments. Imagine that 8 of these apartments are vacant and the landlord's account is currently paying for any usage to these apartments. In addition to changing the landlord agreement for all 100 apartments to the new landlord, the service for the vacant apartments should be transferred to the new landlord. A user can go to premise management, use the search criteria to view all premises for this landlord in a given apartment complex whose service is currently being paid for by the landlord and start service for these premise for the new landlord. (Using the back-to-back start/stop functionality, service will be stopped for the old landlord.)

## Manage Groups of Premises

The powerful search criteria on the premise management page allows you to view different groups of premises based on any combination of account information, landlord information, parent premise address information and general geographic type information. You may also define search criteria such as "premises **not** linked to account 1234567890" or "premises linked to **any** landlord" or "premises **not** linked to any parent premise". For example, you could display all premises in a given latitude / longitude that are not linked to any parent premise.

Once your search results display the desired list of premises, you could perform mass updates such as assigning a parent premise or starting service. Following are examples of situations where this page would be helpful:

- The developer of a high-rise condominium complex is ready to start service for a group of condos. A user uses the premise management search criteria to display the premises linked to a give parent premise (representing the condo complex) that are not linked to any account. The user selects the premises in this list that the developer would like to start and starts service for all premises at once.
- There is a shopping mall in your service territory and each store in an individual premise. You may decide to define a parent premise for the entire shopping mall and link each store's premise to the parent. Once the parent premise is defined, a user navigates to the premise management page and displays the individual premises based on a similar address or based on a similar geographic type. The user then assigns the parent premise for all the individual premises using a single action.

## Premise Management Page

---

This page allows you to display a list of premises using a combination of search criteria. In addition, you can perform actions on one or more of the resulting premises, including:

- Assigning a parent premise to one or more premises

- Removing the link to a parent premise from one or more premises
- Assigning a landlord to one or more premises
- Removing the link to a landlord from one or more premises
- Start service for one or more premises
- Stop service for one or more premises

Open this page using **Customer Information, Premise Management**.

### Description of Page

The top half of the page is where you enter the criteria used to search for premises.

**Multiple search criteria may be specified.** You can search for premises using a combination of search criteria. For example, if you enter both a **Parent Premise Filter** of **Linked To Any Premise** and an **Account Filter** of **Not Linked to Any Account**, only those premises that have a parent premise but are not linked to an account will be displayed.

**At least one positive criterion is required.** You may not define your search criteria to only use filter values that start with "Not Linked...". At least one "positive" filter value is required.

The following table describes each of the different search methods.

Search Method	Description
Landlord Filter	<p>Use this filter to narrow down your search based on landlord. Enter one of the following values:</p> <p><i>Not Applicable</i></p> <p><i>Linked To This Landlord</i> - for this option, you must enter a <b>Landlord Agreement</b></p> <p><i>Not Linked To This Landlord</i> - for this option, you must enter a <b>Landlord Agreement</b></p> <p><i>Linked To Any Landlord</i></p> <p><i>Not Linked To Any Landlord</i>.</p> <p>A filter value of <i>Not Applicable</i> defaults.</p>
Parent Premise Filter	<p>Use this filter to narrow down your search based on parent premise. Enter one of the following values:</p> <p><i>Not Applicable</i></p> <p><i>Linked To This Parent Premise</i> - for this option, you must enter a <b>Premise ID</b> for the parent premise</p> <p><i>Not Linked To This Parent Premise</i> - for this option, you must enter a <b>Premise ID</b> for the parent premise</p> <p><i>Linked To Any Parent Premise</i></p> <p><i>Not Linked To Any Parent Premise</i>.</p> <p>A filter value of <i>Not Applicable</i> defaults.</p>
Account Filter	<p>Use this filter to narrow down your search based on account. Enter one of the following values:</p>

Search Method	Description
	<p><i>Not Applicable</i></p> <p><i>Linked To This Account</i> - for this option, you must enter an Account ID</p> <p><i>Not Linked To This Account</i> - for this option, you must enter an Account ID</p> <p><i>Linked To Any Account</i></p> <p><i>Not Linked To Any Account</i>.</p> <p>A filter value of <i>Not Applicable</i> defaults.</p>
Premise Filter	<p>Use this filter to narrow down your search based on account. Enter one of the following values:</p> <p><i>Located At This Address</i> - for this option, you must enter address constituents</p> <p><i>Located At This Geographic Type / Value</i> - for this option, you must enter a Geo Type and Value</p> <p><i>Not Applicable</i></p> <p><i>Show This Specific Premise</i> - for this option, you must enter a Premise ID</p> <p>A filter value of <i>Not Applicable</i> defaults.</p>

The **Select All / Clear All** buttons are used to select premises if you plan on issuing any of the mass update actions at the bottom of the page. Refer to the description of the “mass update” actions below for more information.

**50 premises at a time.** Clicking **Select All** selects the first 50 premises in the grid. If more than 50 premises exist, you must select them in batches.

The grid that follows contains the premises that match your search criteria. The following information appears in the grid:

- Select box. Use this checkbox to select premises for mass update actions.
- The **Premise Information** column shows information about each premise. Click the hyperlink to transfer to the [Premise](#) page where you can update information about the premise in question.
- The **Account Information** column shows information about all accounts linked to the premise, if any. Only accounts with non-canceled, non-closed service agreements linked to this premise are displayed. The column displays the name of the customer's main person followed by a list of status values for any of the account's service agreements linked to this premise. For example, you may see **Customer A (pending stop), Customer B (pending start)**. If multiple service agreements linked to the account & premise have the same status, the status is only displayed once.
- The **Parent Premise Information** column displays information about the premise's parent premise, if one exists.
- The **Landlord Information** column displays the landlord agreement description for the premise's landlord, if there is one.

This transaction has sophisticated logic that can be used to perform “mass updates” to the premises that appear in the grid using the buttons at the bottom of the page. The buttons are enabled if you select at least one row from the premise grid. The following points describe these mass update actions:

### Contents

- [Start Service](#)
- [Stop Service](#)
- [Assign Parent Premise](#)
- [Remove Parent Premise](#)
- [Assign Landlord](#)
- [Remove Landlord](#)

## Start Service

The **Start Service** button is used to start service for one or more premises. When clicked, the **Start SA Groups Confirmation** window opens.

**Similar to Start/Stop.** This page is similar to the Start SA Confirmation page that appears on start/stop. The differences are to accommodate starting multiple premises.

In order to start service for the selected premises, select the **Account ID**.

The **Start Date** is the date on which the service is scheduled to start.

Use **Start Requested By** to indicate the name of the person who requested the start.

Check **Include Child Premises** if all the child premises in the hierarchy below the selected premises should also be started for this account.

Click **Show SA Groups** to populate the remainder of the page.

The **Service Points** section summarizes the service points that exist for all the chosen premises. A row is displayed for each different **SP Type** along with a **Count** of the number of service points of that type. By default, all rows are checked, assuming that you want to start service for all service points (and the SA Groups To Start scroll below displays proposed SAs to start for each SP type). You may uncheck any SP Type row if you do not wish to start service for that SP type. Click **Show SA Groups** to rebuild the SA Groups To Start information.

**Warning!** This page displays summary information about the service agreements and service points that are being turned on. It is recommended that you start groups of premises that have similar service point configurations. If you attempt to start service for many premises, each with different service point configurations, the start logic may be very complicated to verify on this page.

The **SA Groups To Start** scroll contains an entry for every service agreement that will be created if you click the **Start** button. If you do not want a service agreement to be created, click the – button or uncheck **Start this SA Group**. If you would like to start service agreements that are not SP-based service agreements (for example a deposit or a connection charge), click the + button to add a new SA group. The system assumes that you want to create a new service agreement for each selected premise.

Turn on the **Start this SA Group** switch if you want service agreements of this type to be created. If this is the first customer at the selected premises, this switch is checked for all service agreements that appear in the scroll. For each premise that was linked to a previous customer, this switch is on only for those types of service agreements used by the previous customer(s). (We assume the new customers only want the services used by the previous customers). Service agreements that are possible, but were not used by the previous customers, appear in the scroll with this switch turned off.

If Start This SA Group is checked, the system displays a message indicating the number of service agreements (for the number of service points) that will be created if you choose Start.

For a description of the remaining fields, refer to [Start/Stop - Start Confirmation](#).

Click **Start**. Standard [start](#) logic is followed when clicking start.

**Note.** The start button is disabled if changes are made to fields on the page that may impact the SA Groups To Start collection. You must click **Show SA Groups** to rebuild the SA groups and enable the start button.

## Stop Service

The **Stop Service** button is used to stop service for one or more premises.

**Note** that this functionality stops service for any account linked to the selected premises. If you want to stop service at the selected premises for only certain accounts, use the account filter criteria to narrow down the selection of premises prior to clicking Stop Service.

When clicked, the **Stop SA Group Confirmation** window opens.

In order to start service for the selected premises, enter a **Stop Date**.

Enter a **Restrict to Account** if you want to stop only service agreements linked to this account. If not specified, all active services agreements for the premise are stopped regardless of the account.

Use **Stop Requested By** to indicate the name of the person who requested the stop.

Check **Include Child Premises** if the child premises in the hierarchy below the selected premises should also be stopped.

Click **Show SA Groups** to populate the remainder of the page.

The system builds a collection of service agreements currently linked to the service points for the selected premises. The grid contains the **CIS Division** and **SA Type** for each SA "group" and displays the number of service agreements in the **SA Group Summary**.

By default, the system checks each SA group assuming that you want to stop service for all SAs. You may uncheck any of the SA groups to not stop service for those SAs.

Click **Stop**. Standard [stop](#) logic is followed when clicking stop.

**Note.** The stop button is disabled if changes are made to fields on the page that may impact the SA collection. You must click **Show SA Groups** to rebuild the SA collection and enable the stop button.

## Assign Parent Premise

The **Assign Parent Premise** button is used to assign a parent premise to one or more premises. When clicked, the **Assign Parent Premise** window opens.

Specify the **Parent Premise ID** and click **Assign**.

## Remove Parent Premise

The **Remove Parent Premise** button is used to remove the parent premise from one or more premises. When clicked, the parent premise ID of all selected premises is reset.

## Assign Landlord

The **Assign Landlord** button is used to assign a landlord to one or more premises. When clicked, the **Assign Landlord** window opens.

Specify the **Landlord** agreement ID.

Check **Include Child Premises** if the child premises in the hierarchy below the selected premises should also be assigned this landlord.

Click **Assign**.

## Remove Landlord

The **Remove Landlord** button is used to remove the landlord agreement from one or more premises. When clicked, the **Remove Landlord** window opens.

Check **Include Child Premises** if all the child premises in the hierarchy below the selected premises should also have their landlord agreement removed.

Click **Remove**.

# Umbrella Agreement Management

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...

**Separate module.** Please note that 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.

## Contents

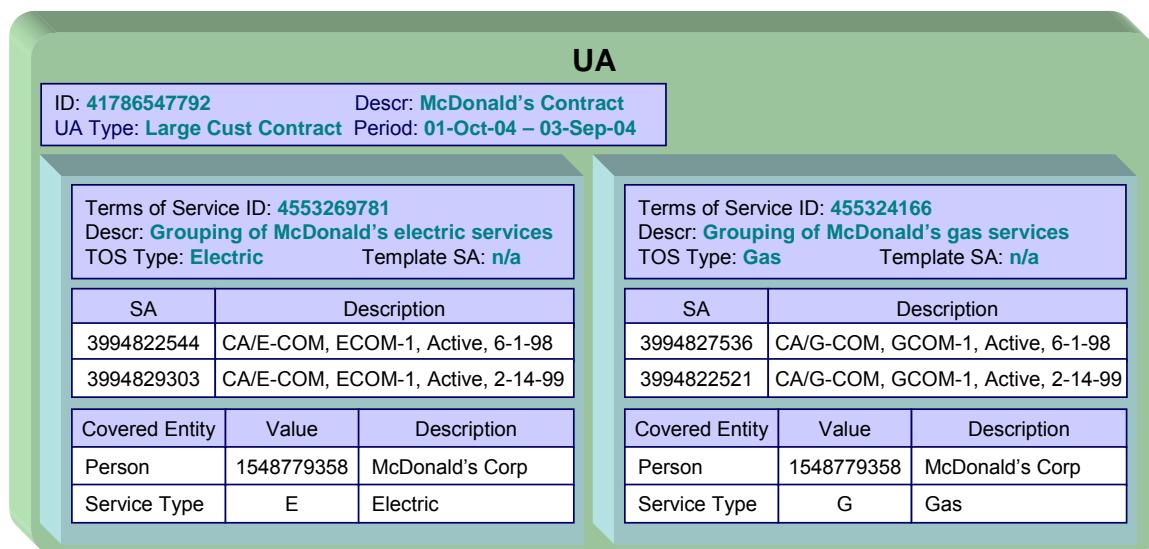
- Overview Of Umbrella Agreement & Terms Of Service
- Umbrella Agreements & Quotes
- Additional Umbrella Agreement Topics
- Umbrella Agreement Maintenance
- Terms Of Service Maintenance

## Overview Of Umbrella Agreement & Terms Of Service

---

An umbrella agreement is used to group together terms for a contract. It contains high level information about the contract such as the effective dates, the renewal date, if applicable and an account management group, if applicable. You may also define characteristics and [contacts](#) for the umbrella agreement.

The terms of the umbrella agreement are defined using one or more terms of service (TOS) records. For each TOS record, you must link one or more service agreements to indicate the services covered under the TOS.



The Terms of Service may point to a ‘template service agreement’ as a way to specify the rating terms and conditions to be used by all service agreements linked to the TOS. If a TOS does not define a template SA, then the TOS is used largely as a grouping mechanism rather than a billing mechanism.

Umbrella agreements and their terms of service records are used to support specific business scenarios that your organization supports.

## Contents

- [Umbrella Agreement Contacts](#)
- [TOS Covered Entities](#)
- [Rate Terms Overridden By Terms Of Service](#)
- [Umbrella Agreement Renewal](#)

## Umbrella Agreement Contacts

When defining an umbrella agreement, you may define one or more [contacts](#) for the umbrella agreement. For example you would create a contact for every person with whom you are working with from the organization negotiating the contract.

A contact must be first defined as a [person](#) in the system.

**Note** that there is no base product functionality that uses this information.

## TOS Covered Entities

Your implementation may configure a TOS type to require that entities “covered” by the terms of service record be referenced on the TOS. For example, imagine that you define an umbrella agreement to provide special discounts to all the McDonalds franchises in your service area. When creating the terms of service record for the umbrella agreement, you may link the person record representing the McDonalds corporate entity as a covered entity for the TOS.

**Note** that there is no base product functionality that uses this collection. However, your implementation may use this information in an algorithm, for example, it could be used in a campaign / package eligibility algorithm.

Refer to [Terms Of Service Covered Entities](#) for information about defining covered entities for a terms of service type.

## Rate Terms Overridden By Terms Of Service

Your organization may require rate information to be temporarily overridden for a service agreement that is linked to a TOS. In order to do this, the TOS record must reference a template SA. Refer to [Overriding Rate Terms](#) for information about defining your TOS types to support template service agreements.

The topics in this section further describe the possible rate term overrides that may occur when a service agreement is linked to a terms of service record.

### Contents

- [Rate Schedule Override by TOS](#)
- [Override Of Other Rate Terms By TOS](#)

### Rate Schedule Override by TOS

If a service agreement is linked to a terms of service, you may configure the system to use the [rate schedule](#) from the terms of service record's template SA rather than the one from the customer's service agreement. The customer's SA type controls this option.

Refer to [Configure SA Types for Umbrella Agreements](#) for more information.

If a service agreement is linked to a terms of service record, the [rate information tree](#) on the service agreement's rate info tab tells the user whether the rate from the service agreement or from the template SA on the TOS is currently applicable.

If a given service agreement's rate schedule is being taken from the terms of service record's template SA, then the override rate schedule is used not just in bill calculations, but for any logic that requires the service agreement's rate schedule. For example, one of the recommended budget amount algorithms may estimate consumption and apply the service agreement's rate. In this case, the rate used by the algorithm is the one linked to the terms of service record's template SA.

### Override Of Other Rate Terms By TOS

There are other terms associated with a rate that may change when linking a service agreement to a terms of service record that has a template SA. These include contract riders, contract terms and tax exemptions.

When applying the rate to create the customer's bill segment, if a bill factor on any of the rate components indicates that it is eligible for tax exemptions, contract terms or contract riders, a flag on the bill factor indicates where the system should look for the appropriate value. The Terms of Service Usage 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.

Refer to [Configure Bill Factors for Umbrella Agreements](#) for more information.

**Independent of rate source.** Note that the flag values on the bill factors are independent of the rate source setting for the SA type. For example, your implementation may configure the SA type's rate source to indicate that rate should be taken from the TOS template SA. However, if a bill factor is eligible for tax exemptions, your implementation may configure the bill factor to indicate that the exemption information should be taken from the customer's SA, not the template SA.

If a service agreement is linked to a terms of service record, the [rate information tree](#) on the service agreement's rate info tab tells the user the

## Umbrella Agreement Renewal

Although umbrella agreement records require an end date, your organization probably allows for renewal of agreements. A renewal may involve full re-negotiation of all terms, or may be as simple as extending the current terms. If umbrella agreements of a given type are eligible to be renewed, you configure the UA type to indicate how to calculate the renewal date and which algorithm to execute when the renewal date is reached.

### Umbrella Agreement Renewal Background Process

The UA renewal background process (known by the batch control ID of [UARENEW](#)) finds **complete** umbrella agreements whose renewal date has arrived and executes the renewal algorithm for the associated UA type.

**Note.** It is the responsibility of the renewal algorithm to update the renewal date as per the business practice. Refer to [Renewable Umbrella Agreements](#) for more information.

## Umbrella Agreements & Proposals

Umbrella agreements and their terms of service records may reference proposal service agreements. This functionality allows you to create the full contract for your customer during the quoting stage.

## Additional Umbrella Agreement Topics

The following topics describe additional functionality related to umbrella agreements.

### Contents

- [Umbrella Agreement Alerts](#)
- [Orders And Umbrella Agreements](#)

### [Umbrella Agreement Reports](#)

## **Umbrella Agreement Alerts**

The [alert zone](#) will highlight if any of the account's SAs are linked to an effective umbrella agreement (via a non-canceled terms of service record).

## **Orders And Umbrella Agreements**

When starting new service for a given account and premise, you may know ahead of time that the new service should be linked to an existing umbrella agreement. For example, perhaps all the service in a specific low income housing area qualifies for a special negotiated contract for the next year.

The system provides algorithms to allow you to add a new service agreement to an existing terms of service record when using the [Order](#) transaction to start service.

Refer to [Configure A Campaign To Link An SA To A Terms Of Service](#) for more information.

## **Umbrella Agreement Reports**

The system supplies a [Report](#) providing umbrella agreement information for one or more umbrella agreements.

## **Umbrella Agreement Maintenance**

---

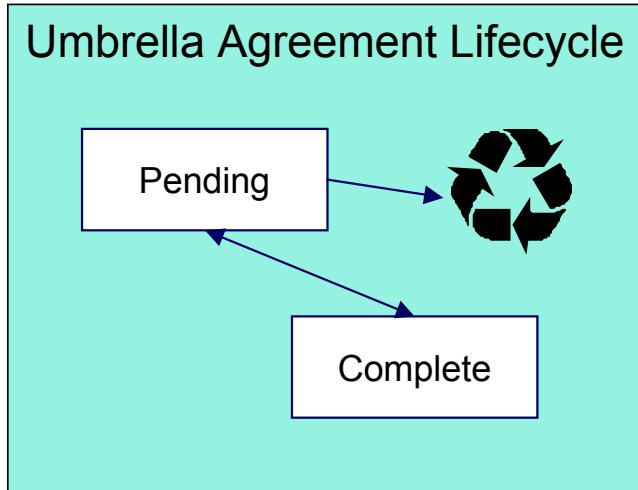
Umbrella agreements help you manage your contracts by allowing you to group together service agreements for the contract with one or more terms of service records for the umbrella agreement.

### **Contents**

- [Lifecycle Of An Umbrella Agreement](#)
- [Umbrella Agreement - Main](#)
- [Umbrella Agreement - Contacts](#)
- [Umbrella Agreement - Characteristics](#)

## **Lifecycle Of An Umbrella Agreement**

The following diagram illustrates the lifecycle of an umbrella agreement.

**Pending**

The umbrella agreement is created in ***pending*** status.

You may delete a ***pending*** umbrella agreement.  
Deleting a UA also deletes its terms of service records.

**Complete**

Only ***complete*** umbrella agreements are considered effective. You may transition a UA to complete if there is at least one terms of service record linked to it and if none of its TOS records are in ***pending*** status.

Under unusual circumstances, a ***complete*** umbrella agreement may be changed back to ***pending*** to correct errors.

## Umbrella Agreement - Main

Open this page using **Umbrella Agreement Management**, **Umbrella Agreement**.

### Description of Page

**Umbrella Agreement Info** contains a concatenation of important information about the umbrella agreement. **Umbrella Agreement ID** is the system-assigned unique identifier of the umbrella agreement. These values only appear after the UA is added to the database.

**Umbrella Agreement Status** shows the current state of the UA. The buttons adjacent to the status allow you to transition the UA's status.

- Click **Complete** to change the status of a ***pending*** umbrella agreement to ***complete***.
- Click **Reopen** to change the status of a ***complete*** umbrella agreement back to ***pending***.

Refer to [Lifecycle of an Umbrella Agreement](#) for more information.

Enter a **Description** for the umbrella agreement.

The **Umbrella Agreement Type** defines the type of umbrella agreement. Refer to [Setting Up Umbrella Agreement Types](#) for more information.

The **Start Date** and **End Date** define the effective dates of the contract. If the UA type is **renewable**, the **Renewal Date** indicates the date the umbrella agreement is processed by the [renewal background process](#).

You may assign an **Account Management Group** to this umbrella agreement. The AMG controls the roles assigned to [To Do entries](#) associated with the umbrella agreement. Refer to [Setting Up Account Management Groups](#) for more information.

Use the **Comments** area to describe anything interesting or unusual about the umbrella agreement.

This bottom of this page contains a [tree](#) that shows the various objects linked to the umbrella agreement. You can use this tree to both view high-level information about these objects and to transfer to the respective page in which an object is maintained.

## Umbrella Agreement - Contacts

The contacts page defines persons associated with the umbrella agreement. Open this page using **Umbrella Agreement Management**, **Umbrella Agreement** and navigate to the **Contacts** tab.

### Description of Page

The contacts collection allows you to define persons that are associated with the umbrella agreement. For each contact, define the **Person ID** for the person record that contains the demographic information for the contact. For each contact, use the **Comments** to define information about the contact.

## Umbrella Agreement - Characteristics

The characteristics page contains information that describes miscellaneous information about the umbrella agreement. Open this page using **Umbrella Agreement Management**, **Umbrella Agreement** and navigate to the **Characteristics** tab.

### Description of Page

**Characteristic Types.** You can only choose characteristic types defined on umbrella agreement's [umbrella agreement type](#).

The following fields display:

**Characteristic Type**      Indicate the type of characteristic.

**Characteristic Value**      Indicate the value of the characteristic.

**Default Note.** An umbrella agreement's characteristics default from the [umbrella agreement type](#).

## Terms Of Service Maintenance

A terms of service (TOS) record defines one or more service agreements participating in a given umbrella agreement. The TOS record may also reference a template SA that could be used to override rate terms for the participating service agreements.

### Contents

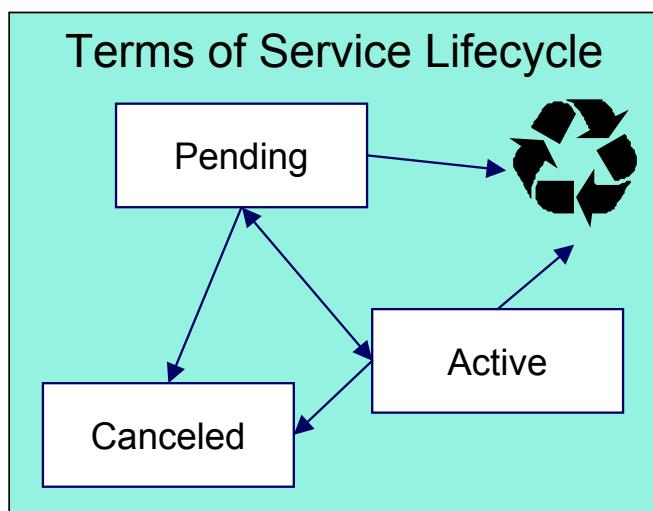
[Lifecycle Of Terms Of Service](#)

[Terms Of Service - Main](#)

[Terms Of Service - SAs](#)

## Lifecycle Of Terms Of Service

The following diagram illustrates the lifecycle of a terms of service record.



### Pending

The terms of service record is created in **pending** status. You may delete a **pending** TOS. If you want to keep a record of the TOS, you may cancel it rather than deleting it.

### Active

Only **active** terms of service records are considered effective. Change the status to **active** when you have entered all the desired information for this TOS. You may delete an active terms of service if the umbrella agreement is still **pending**. (Otherwise you must cancel the TOS.)

### Canceled

You may **cancel** a pending or active terms of service if it should no longer be used. Canceling a terms of service record unlinks all its service agreements.

## Terms Of Service - Main

Open this page using **Umbrella Agreement Management, Terms of Service**.

### Description of Page

**Terms of Service Info** contains a concatenation of important information about the terms of service record. **Terms of Service ID** is the system-assigned unique identifier of the terms of service record. These values only appear after the TOS is added to the database.

The **Umbrella Agreement ID** defines the umbrella agreement to which this terms of service is linked.

**Status** shows the current state of the terms of service record. The buttons adjacent to the status allow you to transition the TOS record's status.

- Click **Activate** to change the status of a *pending* terms of service record to *active*.
- Click **Pend** to change the status of an *active* terms of service record back to *pending*.
- Click **Cancel** to change the status of a *pending* or *active* terms of service record to *canceled*.

Refer to [Lifecycle Of Terms of Service](#) for more information.

Enter a **Description** for the terms of service record.

The **Terms of Service Type** defines the type of terms of service record. Refer to [Setting Up Terms of Service Types](#) for more information.

Use the **Comments** area to describe anything interesting or unusual about the umbrella agreement.

The **Characteristics** collection contains information that describes miscellaneous information about the terms of service record.

**Characteristic Types.** You can only choose characteristic types defined on the terms of service record's [terms of service type](#).

The following fields display:

**Characteristic Type**      Indicate the type of characteristic.

**Characteristic Value**      Indicate the value of the characteristic.

**Default Note.** A terms of service record's characteristics default from the [terms of service type](#).

The **Covered Entities** collection allows you to define specific entities that are "covered" by this terms of service record. Refer to [TOS Covered Entities](#) for more information.

**Characteristic Types.** You can only choose covered entity characteristic types defined on the terms of service record's [terms of service type](#).

The following fields display:

<b>Covered Entity Characteristic Type</b>	Indicate the type of characteristic used to define this covered entity type.
<b>Characteristic Value</b>	Indicate the value of the characteristic used to define the covered entity value.

**Default Note.** A terms of service record's covered entity types default from the [terms of service type](#).

## Terms Of Service - SAs

This page is used to maintain the service agreements linked to the terms of service record.

Open this page using **Umbrella Agreement Management**, **Terms of Service** and navigate to the SAs tab.

### Description of Page

**Terms of Service Info** contains a concatenation of important information about the terms of service record. **Terms of Service ID** is the system-assigned unique identifier of the terms of service record.

The filters control the service agreements that appear in the grid. The following points describe the various options:

- Use the **SA Filter** to define the types of service agreements that appear in the grid. The following options are available:
  - **Account.** Use this option to restrict service agreements to those linked to a given **Account ID**.
  - **Address.** Use this option to restrict service agreements to those linked to service points associated with a given **Address**, **City** and/or **Postal** code. Note, you can specify any combination of these fields.
  - **All.** Use this option if you do not wish to restrict service agreements based on service agreement attributes.
  - **All Proposals.** Use this option to restrict service agreements to only proposal service agreements.
  - **All Real SAs.** Use this option to restrict service agreements to only “real” service agreements (i.e., not proposal SAs).
  - **Geographic Type.** Use this option to restrict service agreements to those linked to service points associated with a given **Geographic Type** and **Geographic Value**.
  - **SA Type.** Use this option to restrict service agreements to those linked to a given **CIS Division** and **SA Type**.
  - **Specific SA.** Use this option to reference a specific **SA ID**.

- Use **Status Filter** to restrict the service agreements based on their status. Status values for both proposal service agreements and regular service agreements are included in the filter. The following options are available:
  - **All.** This option shows all service agreements, including proposal service agreements, regardless of status.
  - Refer to [Service Agreement Lifecycle](#) for the various status values for real service agreements.
  - Refer to [Proposal SA State Transition](#) for more information about status values for proposal service agreements.

Don't forget to click the search button after changing the filters.

The grid that follows contains the **Service Agreements** that match your search criteria along with its **Account Info**.

Use the "+" to link new service agreements to the terms of service record.

**Link SA Using Order.** You may also [link a service agreement to a TOS through orders](#) if you have configured an appropriate campaign and package.

Note you may remove a service agreement from the TOS using the “-“ button as long as the change has not been committed to the database. As soon as the SA's link to the TOS has been committed to the database, you must use the **Remove** action to unlink the service agreement.

The **Select All / Clear All** buttons are used to select service agreements if you plan on issuing the mass update action **Remove** at the bottom of the page. Clicking **Remove** removes the service agreement from the terms of service record.

# Job Streams

The term "job stream" refers to a suite of batch jobs that run periodically. Most organizations have multiple job streams. For example,

- You'll have a job stream that contains batch jobs that run nightly
- You'll have a different job stream that contains the hourly batch jobs
- You'll have a different job stream that contains the batch jobs that extract data for your data warehouse
- Etc.

The topics in this section describe how to execute and monitor your job streams.

Refer to [The Big Picture of Scheduling Batch Jobs](#) for a description of how to set up the system to enable this functionality.

**Only used by some organizations.** Many organizations use a third party tool to manage the periodic execution of batch jobs rather than the system batch scheduler. If this is true for your organization, this section is not applicable.

**Separate module.** Please note that the batch scheduler functionality is associated with a separate [Workflow Scheduling](#) 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. .

## Contents

- [Job Stream Summary](#)
- [Job Stream Details](#)

## Job Stream Summary

This page presents a summary of your job streams. Open this page using **Batch, Job Stream Summary**.

### Description of Page

This page summarizes the status of your job streams.

**Note.** Only job streams defined on the [Feature Configuration](#) for the Batch Scheduler appear on this page.

The following information is displayed for each job stream:

- **Job Stream.** This is the description of the job stream. Click the hyperlink to transfer to the job stream's workflow process template (the workflow process template defines the number and type of batch jobs in the job stream as well as their dependencies).

- **Cancel Button.** If the job stream is executing, a **Cancel** button appears. Pushing this button will cancel the execution of any batch jobs in the job stream that have not already started. Batch jobs that are currently executing will continue to execute until they conclude or are stopped in the operating system.
- **Start Button.** If the job stream is not executing, a **Start** button appears. Pushing this button submits the request to execute the job stream's batch jobs. The job stream's batch jobs will only execute when the batch submitter next runs. The batch submitter's frequency of execution is configured by your organization.
- **Last Completed Run.** This is the start date / time when the job stream last executed. In addition, the job stream's total elapsed time is displayed. Click the hyperlink to transfer to the [Job Stream Details](#) page where you can see the execution status of each batch job in the job stream.
- **Run In Progress.** This column only contains information if a job stream is executing. The date / time that the job stream was requested is shown as well as a colorful status bar. Clicking the hyperlink under the date / time transfers you to the [Job Stream Details](#) page where you can see the execution status of the job stream's batch jobs.

The status bar provides a visual clue as to the execution status of the job stream's batch jobs. The bar's segments and their colors are used to highlight the relative number of batch jobs in a given state. For example, it will highlight the relative number of **Pending** versus **Complete** versus **Error** batch jobs.

You can hover the mouse over a section of the bar to see the number of batch jobs in each state. You can click on a segment of the bar to drill down to the [Job Stream Details](#) page. When this page appears, only the batch jobs in the respective state are displayed.

The message **Other(s) running** appears if more than one instance of this job stream is running (this should be rare).

- **Next Scheduled Run.** This is the start date / time when the job stream is next scheduled to run. Click the hyperlink to transfer to the [Job Stream Creation Schedule](#) page where you can define when the job stream runs. You can also use this transaction to hold prescheduled runs.

## Job Stream Details

---

Job streams execute many times over their lifetime. For example, the nightly job stream executes every night; the hourly job stream executes hourly; the data warehouse extract executes weekly; etc.

Every time a job stream executes, a workflow process is created. The workflow process's events cause the job stream's batch jobs to be submitted.

This page shows the status of the job stream's workflow events and the batch jobs that they submit. Open this page using **Batch, Job Stream Details**.

### Description of Page

**Job Stream** is a concatenation of important information about the execution of the job stream (e.g., its description, status, start date).

**Job Stream ID** is the system-assigned unique identifier of an execution of the job stream.

**Note.** The Job Stream ID is the unique identifier of the workflow process used to manage the execution of the job stream's batch jobs.

The colorful status bar summarizes the execution status of the job stream's batch jobs. This bar's segments and their colors are used to highlight the relative number of batch jobs in a given state. For example, it can highlight the relative number of **Pending** versus **Complete** batch jobs (**Pending** jobs are one color, **Complete** jobs another).

You can hover the mouse over a section of the bar to see the number of batch jobs in each state. You can click on a segment of the bar to cause the grid to be refreshed with batch jobs in the respective state. For example, if you click on the color used to highlight jobs in **Error**, only these jobs will appear in the grid. You can achieve the same results by selecting the desired **Event Status**.

If the job stream is executing, a **Cancel Job Stream** button appears. Pushing this button will cancel the execution of any batch jobs in the stream that have not already started. Batch jobs that are currently executing will continue to execute until they conclude or are stopped in the operating system.

The grid contains a summary of the job stream's workflow events and the batch jobs that they submit. The following information is displayed for each workflow event:

- **Event** describes the workflow event that submits the batch job (the code following the description is the workflow event's type).  
The colorful box that prefixes the event provides a visual clue of the event's status (e.g., **Pending** events are one color, **Waiting** another, etc.). These colors correspond with the "status bar" at the top of the page.  
Click on the hyperlink to transfer to the [workflow event](#). You might need to display a workflow event if:
  - You need to cancel a **Pending** or **Waiting** event. To do this, display the event and change its status to **Canceled**. Note, if the **Waiting** event is associated with a batch job that is currently executing, canceling the event will NOT abort the execution of the batch job. The only way to abort the execution of a batch job that is executing is via the operating system.
  - You need to restart a batch job that's in **Error**. To do this, display the workflow event and change its status to **Pending**.
- **Batch Run** provides an overview of the batch job submitted by the workflow event:
  - If the corresponding workflow event is **Pending**, the batch job has not been submitted yet. There are two reasons why an event may be **Pending**:
    - If the event is waiting for the batch submitter to run, the message **Starting soon...** appears. The next time the batch submitter runs, the batch run will be executed. The batch submitter's [frequency of execution](#) is configured by your organization.
    - If the event is waiting for the successful completion of other events, a message describing the events on which it is waiting is shown. You define the dependencies between events (i.e., batch jobs) when you set up the job stream's workflow process template.
  - If the corresponding event is **Waiting**, the batch job is currently executing and a summary of the execution status is shown. You can click on the hyperlink to transfer to the Batch Run Tree where the details of the run's parallel threads and errors are shown. Note, a summary of the various threads and errors is shown in the next column.

- If the corresponding event is **Complete**, **Failed**, or **Error**, the batch job has completed and a summary of its execution is shown. You can click on the hyperlink to transfer to the [Batch Run Tree](#) where the details of the run's parallel threads and errors are shown.
- **Batch Run Details** provides more extensive details about a batch run's execution. The following information is shown:
  - You can configure the workflow event to execute multiple parallel threads of a given batch job. For example, rather than running a single instance of the BILLING batch job, you can indicate 40 parallel threads should run (thus improving throughput). This column always contains a summary of the execution status of the parallel threads.
  - The number of records processed by the job. For example, if this is the BILLING job, the number of records shows the number of accounts for which bills were created.
  - The number of errors detected during the batch job (this information appears in red).
  - The average, minimum and maximum execution time per thread.

You can click on the hyperlink in the adjacent column to transfer to the [Batch Run Tree](#) where the details of the run's parallel threads and errors are shown.

# Workflow and Notifications

We use the term “notification” to reference the electronic transactions that you exchange with third parties when:

- They need information about a customer
- They need to change something about a customer

For example, an energy service provider sends a notification to a distribution company when a customer chooses them as their energy provider. When a notification is received by the distribution company, the system responds by creating a workflow process. The workflow process contains workflow events. These events perform the processing necessary to execute the notification.

In addition to processing incoming notifications, workflow processes can be used to automate many different types of multi-event processes. For example, you can use workflow processing to manage the events associated with the inspection of a new premise.

This section describes how to manage your workflow and notification processing.

In the section, [Defining Workflow and Notification Options](#), we describe how to set up the control tables that automate most of your notification and workflow processing. If you set these tables up properly, little or no operator intervention should be necessary. You should only have to access the workflow and notification pages when you need to override the standard procedures embodied in your workflow processes.

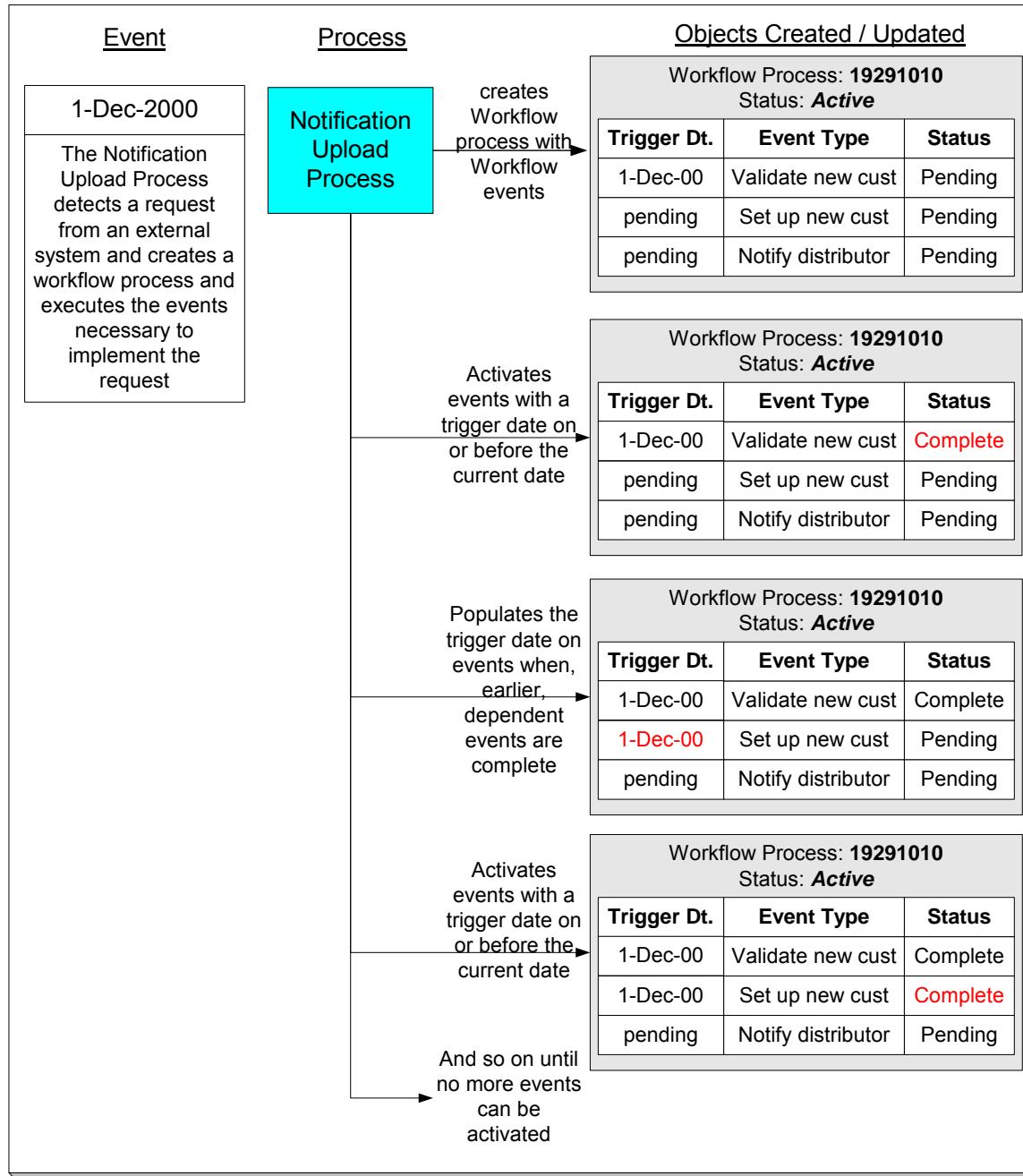
## Contents

- [The Lifecycle Of A Workflow Process And Its Events](#)
- [Workflow Process Maintenance](#)
- [Workflow Event Exception](#)
- [Uploading Notifications](#)
- [Downloading Notifications](#)

## The Lifecycle Of A Workflow Process And Its Events

---

The following diagram illustrates the lifecycle of a typical workflow process and its workflow events:



The above diagram is referenced in the following topics.

## Contents

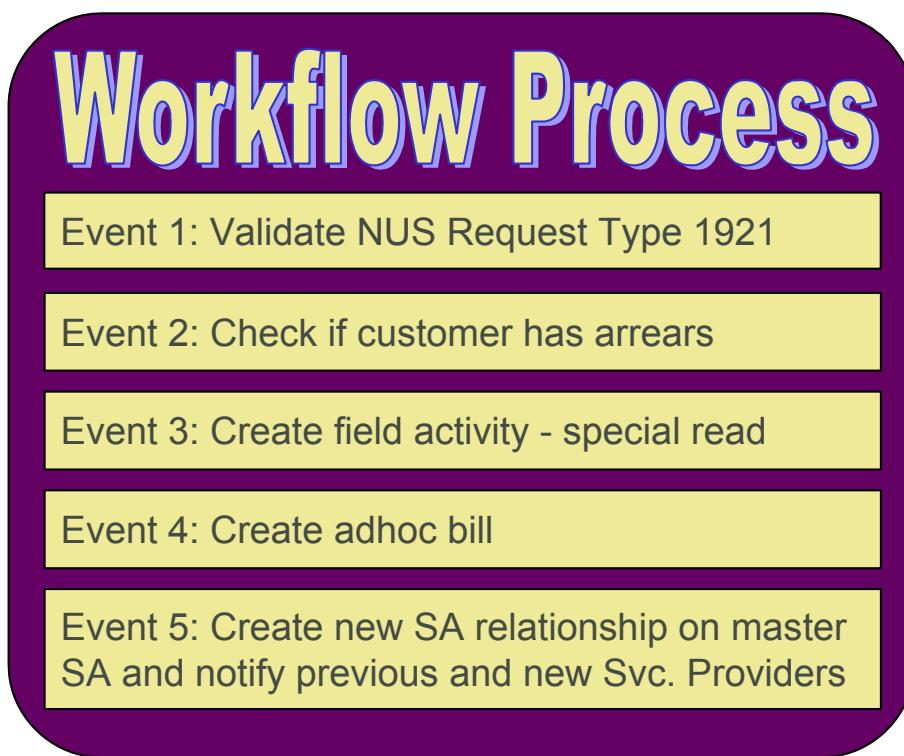
- [What Is A Workflow Process?](#)
- [A Workflow Process Contains Relevant Data](#)
- [How Are Workflow Processes Created?](#)
- [How Are Workflow Events Completed?](#)
- [How Are Workflow Processes \(and their Events\) Canceled?](#)

## What Is A Workflow Process?

A workflow process is difficult to define because its flexible design can be used to automate many different types of multi-event processes. For example,

- You can use a workflow process to manage the events associated with the inspection of a new premise.
- You can use a workflow process to manage the events that transpire in a deregulated market when a customer wants to switch energy service providers.
- You can use a workflow process to notify interested parties when something about a customer changes. For example, if a customer changes their phone number, a workflow process can send notifications to the customer's various service providers.

The following is an illustration of a workflow process that could be used to process a request from an energy service provider to switch a customer's electric supplier on a specified date.



**Bottom line:** A workflow process contains one or more workflow events. Each event is activated on its respective trigger date. When an event is activated, the system executes the activation algorithm associated with the event's event type. It's up to you to define exactly what algorithms should be associated with your workflow event types. Refer to [Designing Workflow Event Types](#) for sample events.

## A Workflow Process Contains Relevant Data

Linked to a workflow process are the relevant data associated with the process. This information can be for audit purposes or it may be needed by one or more of the event algorithms linked to the process to do its work. For example,

- If a workflow process is associated with the inspection of a premise, the premise is linked to the workflow process.
- If a workflow process is associated with new service, many pieces of information are needed to be linked to the workflow including the address of the new service location, the date service should start, perhaps the desired rate and perhaps customer information, if this is for a new customer.
- If a workflow process is triggered when we receive a notification that we have a new customer, the notification is linked to the workflow process.

This data may be linked to the workflow process in one of two ways: as context or as characteristics. The following points describe each method:

- Context Collection
  - Uses a context type to identify the type of data. The values of the context type are stored in a lookup field and must be customized by your implementers.
  - The values are stored in the context value field. Values may be up to 50 bytes long.
  - The primary key of the record includes context type and context value so multiple values may be entered for the same type.
  - Note, the context value is a generic field. If the value of the field is a foreign key, there is no way for the system to enable drill down capability to this foreign key's entity.
- Characteristic Collection
  - Uses the [characteristic type](#) table to identify the type of data. Characteristic types may be pre-defined, ad-hoc, foreign key or file location.
  - The values are stored in different characteristic value fields depending on the type of characteristic. The ad-hoc and file location values may be up to 254 bytes long.
  - The primary key of the record includes characteristic type and a sequence number so multiple values may be entered for the same type.
  - If the characteristic is of type foreign key reference, a search is available for selecting an appropriate value and a drill down key is available for viewing the related record.

**Note.** The original design included only the context collection. The characteristic collection was added later as a result of limitations in the context functionality that was supported in other parts of the system by characteristic functionality. It is expected that characteristics will be the preferred method to use for implementations. Context will continue to be supported for earlier releases.

Information in a workflow process' context or characteristics collection is referenced by the workflow process' event algorithms when they are activated.

If a workflow process is created to process an incoming notification from a third party, the first event algorithm for your workflow process must populate the workflow process' context or characteristics with individual fields specified on the incoming notification.

Refer to [Uploading Notifications Into The System](#) for more information.

**Note.** If an account ID or premise ID in Oracle Utilities Customer Care and Billing is linked to an **active** workflow process either via context or via a characteristic, you may configure your system to display an alert when that account or premise is displayed in the current context. To display this alert, create one or more algorithms for the algorithm type [CCAL-WF](#), setting the parameters as appropriate to display account or premise and indicate whether or not the account/premise will be found in the context or characteristic collection. Plug in your new algorithm(s) to the [installation option – algorithm](#) collection.

## How Are Workflow Processes Created?

A workflow process may be created as follows:

- The Notification Upload background process (referred to by the batch code of [WFPRINIT](#)) creates a workflow process when a notification is interfaced into the system. The workflow process has one or more workflow event(s). Workflow processes created as a result of a notification upload staging record will reference the NUS ID that created it. Refer to [The Big Picture Of Notification Processing](#) for more information about notifications.
- An operator may create an ad hoc workflow process at will.
- If you customize the system accordingly, the system will create a workflow process when a significant event occurs. The purpose of this type of workflow process is to send outgoing notifications to other interested parties to apprise them of the event. For example, assume you are a distribution company, you probably have to advise a customer's energy supply company whenever their meter is exchanged. In this example, when a meter exchange transpires, you'd want to create a workflow process that would, in turn, send outgoing notifications to the energy supply company. Refer to [Designing Notification Downloads](#) for more information.

Whenever a workflow process is created, the system uses a workflow process *template* to create the workflow process. The workflow process template defines the various events that will transpire over the course of the process. Refer to [Designing Workflow Process Templates](#) for more information.

In the above diagram, the event on 1-Dec-00 shows the creation of a workflow process with three workflow events. Notice that the events all exist in the **Pending** state. The events will exist in this state until their trigger date is on or before the current date.

Refer to [What Type Of Workflow Process Is Created?](#) to understand how the Notification Upload Process creates workflow processes when notification requests are interfaced into the system. Refer to [The Big Picture Of Workflow Events](#) for more information about a workflow process's events.

## How Are Workflow Events Completed?

Several background processes can cause a workflow event to complete as described below:

- A background process (referred to as [WFET](#)) activates events whose trigger date is no longer in the future. (An event's trigger date can be set to a future date when its earlier, dependent events are completed if the workflow process template so indicates). After it activates an event, this process also activates later dependent events until there are no more events to activate. Refer to [Workflow Event Dependencies & Trigger Date](#) for more information.

**Note.** WFET attempts to activate workflow events for a workflow process. The system orders the workflow process records by their workflow process ID, which is a system-generated random number. As a result, you should ensure that the activation algorithms do not expect WF process records to be processed in any other order, such as the order that the WF process records were created. For example, if a “drop request” and an “add request” are both received for the same customer, two workflow processes are created. The activation algorithms for the events for these two processes should work regardless of whether the “drop request” is processed first or the “add request” is processed first.

- The various waiting background processes are responsible for activating events when the thing they are waiting on is complete. For example, if an event creates a pending field activity, it goes into the **Waiting** state. The **WAITFA** background process will activate the **Waiting** event when the field activity becomes **Complete**. Refer to [Waiting Events And Their Waiting Process](#) for more information.

In addition, if the [workflow event type](#) indicates that manual completion is allowed, a user may change the status of the event to **complete** using the [workflow process](#) user interface. When an event is manually completed, the system also activates later dependent events until there are no more events to activate.

Refer to [The Big Picture Of Workflow Events](#) for more information about the various types of workflow events and what happens when they are completed.

## How Are Workflow Processes (and their Events) Canceled?

The background process that is responsible for activating events (referred to as **WFET**) automatically cancels a workflow process when one of its events fails. Refer to [Some Workflow Events May Fail](#) for more information.

Refer to [How Are Workflow Events Canceled](#) for more information.

Besides these automatic cancellations, an operator may cancel a workflow process at will.

## Workflow Process Maintenance

A workflow process is hard to define because its flexible design can be used to automate many different types of multi-event processes. For example,

- You can use a workflow process to manage the events associated with the inspection of a new premise.
- You can use a workflow process to manage the events that transpire in a deregulated market when a customer wants to switch energy service providers.

For more information about workflow processes, refer to [The Lifecycle Of A Workflow Process And Its Events](#).

The topics in this section describe the pages on which workflow process-related information is maintained.

### Contents

- [Workflow Process - Main](#)
- [Workflow Process - Char/Context](#)
- [Workflow Process - Events](#)
- [Workflow Process - Event Char/Context](#)

## Workflow Process - Main

The Main page contains core workflow process information. Open this page using **Workflow and Notification, Workflow Process**.

### Description of Page

The top portion of every page contains the following information:

**Workflow Process** is a concatenation of summary information about this workflow process. It is composed of the workflow process template's name and description, the workflow process's status, and the creation date of the process.

**Workflow Process ID** is the unique system-generated identifier for this workflow process.

The remaining fields are maintained on the main workflow process page.

The **Create Information** describes how the workflow process was created. If the workflow process was created by the system, the identity of the process, and related control table references are displayed. If the workflow process was created by a user, the message indicates this.

**Workflow Status** defines the state of the workflow process. The following values may exist:

**Active**      The system creates a workflow process in the **Active** state. A workflow process remains in this state until the last event linked to the process is complete OR an event fails OR you want to cancel the process for whatever reason.

**Inactive**      A workflow process becomes **Inactive** when the last event linked to the process is completed OR an event fails OR you want to cancel the process for whatever reason. The **Status Reason** field tells you how a workflow process became inactive.

To completely understand the significance of a workflow process's status, refer to [The Lifecycle Of A Workflow Process And Its Events](#).

**Status Reason** defines how a workflow process became **Inactive** or why a workflow process is **Active**. The following values may exist:

**Canceled by User**      This reason should be used when an operator manually inactivates a workflow process (for whatever reason).

<b>Canceled by System</b>	This reason is populated by the system when it cancels a workflow process.
<b>Events Pending</b>	This reason is populated by the system when a process has pending events.
<b>Complete by System</b>	This reason is populated by the system when it completes the last event linked to the workflow process.
<b>Complete by User</b>	This reason is populated by an operator when they manually complete a workflow process.

**Workflow Process Template** defines the workflow events in the process. You can override these events on the **Events** page.

**Changing the template.** If you change the template when all events are **pending**, the system will remove the **pending** events and replace them with the new template's events.

For more information, refer to [Setting Up Workflow Process Templates](#).

If your event has been canceled by the system, a **Message** appears indicating the reason for cancellation.

**Create Date/Time** defines the start date/time of the workflow process. This is important as it affects the trigger dates of the process's workflow events. If you change this date after events exist, the system will NOT update the trigger date on the events. If you need to change the trigger date on the events, you can do this manually on the **Events** page OR you can change the Workflow Process Template to something (anything) and then change it back to the desired template. When you change the template, the system removes the events and creates new ones using the creation date / time on the process.

The workflow process was created by a **Notification Upload** staging record, information about the [notification upload staging](#) record is displayed.

Enter any **Comments** about the workflow process.

This bottom of this page contains a [tree](#) that shows the various objects linked to the workflow process. You can use this tree to both view high-level information about these objects and to transfer to the respective page in which an object is maintained.

## Workflow Process - Char/Context

The Process Char/Context page lists the context collection and/or the characteristic collection associated with the workflow process. Open this page using **Workflow and Notification**, **Workflow Process**, **Process Char/Context**.

Refer to [A Workflow Process Contains Relevant Data](#) for more information about context and characteristics.

### Description of Page

Use characteristics and/or context to capture data to be used by your workflow process' event algorithms. Refer to [A Workflow Process Contains Relevant Data](#) for more information.

The **Characteristics** grid displays the **Characteristic Type** and **Characteristic Value** for each attribute that can be accessed by the workflow process' event algorithms.

The **Context** grid displays the Context Type and Context Value for each attribute that can be accessed by the workflow process's event algorithms.

**Note.** The values for Context Type are customizable using the Lookup table. This field name is WF\_PROC\_CTXT\_FLG.

## Workflow Process - Events

The events page contains the activities that will be performed over the course of the workflow process. Open this page using **Workflow and Notification**, **Workflow Process**, **Events**.

We strongly recommend that you understand the information described in [How Are Workflow Events Completed?](#) before using this page.

The number and type of events associated with a workflow process are defined by the process's workflow process template (which is defined on the first page). Refer to [Setting Up Workflow Process Templates](#) for more information.

### Description of Page

The top portion of every page in this page contains the following information:

**Workflow Process** is a concatenation of summary information about this workflow process. It is composed of the workflow process template's name and description, the workflow process's status, and the creation date of the process.

**Workflow Process ID** is the unique system-generated identifier for this workflow process.

The scroll controls the display of the individual workflow events that comprise the workflow process. The grid shows the events that must be completed before the workflow event is triggered. Only one workflow event is displayed at a time.

**Event Sequence** is the unique identifier of the event.

**Workflow Event Status** defines the state of the event.

For more information, refer to [Workflow Event Lifecycle](#).

If the workflow event status is **Error**, then the message related to the error is displayed adjacent.

**Failed events are special.** If the workflow event's status is **Failed**, the system displays the **Failure Algorithm** that was executed (the type of failure algorithm is defined on the event's event type) and, if the failure algorithm created a workflow process, the system displays information about the **Created Workflow Process**.

The **Workflow Event Type** defines the event's activity.

For more information, refer to [Setting Up Workflow Event Types](#).

If **Dependent on Other Events** is turned on, then the event's trigger date is only determined after the event(s) on which it depends are completed. When this switch is on, a grid appears at the bottom of the page in which you define the event(s) on which this event depends.

For more information about dependent events, see [Workflow Event Dependencies & Trigger Date](#).

The **Trigger Date** defines the date when the system completes (i.e., executes) the event. This field is gray when the event is dependent on the completion of other events.

For more information about a workflow event's trigger date, see [Workflow Event Dependencies & Trigger Date](#).

**Days After Previous Response** defines how a dependent event's trigger date is calculated. The trigger date is set equal to **X** days after the last dependent event is completed, where **X** is the number in this field. This field is gray when the event is not dependent on other events.

For more information about dependent events, see [Workflow Event Dependencies & Trigger Date](#).

The **Status Change Date/Time** is date and time the event's status was last changed.

If the event can go into the **Waiting** state, the **Wait Date/Time** is populated with the date / time on which the event started waiting.

Refer to [Waiting Events And Their Waiting Process](#) for more information.

The grid at the bottom of the page contains the events on which this event depends. This grid is only visible when the event is dependent on other events (as defined by the Dependent on Other Events switch).

## Workflow Process - Event Char/Context

The Event Context page lists the entity(ies) that were created as a result of the execution of a workflow event. For example, if an event creates a field activity, the field activity would be displayed on this page. The event algorithm may choose to record the entity using a characteristic or a context entry. Open this page using **Workflow and Notification, Workflow Process, Event Char/Context**.

Refer to [A Workflow Process Contains Relevant Data](#) for more information about context and characteristics.

### Description of Page

Your event algorithms may record information that was created by the algorithm. The information may be recorded as a characteristic or in a context entry. This information is for audit purposes only.

If **Characteristics** are used to record entities created by the algorithm, this grid displays the **Characteristic Type** and **Characteristic Value** for each entity that was created as a result of the execution of workflow event.

If **Context** entries are used to record entities created by the algorithm, this grid displays the **Context Type** and **Context Value** for each entity that was created as a result of the execution of workflow event.

**Note.** The values for Context Type are customizable using the Lookup table. This field name is WF\_EVT\_CTXT\_FLG.

## Workflow Event Exception

As described in [Workflow Events May Be In Error](#), a record is written the workflow event exception record for every workflow event record that is in error.

To view the messages associated with the exception records, schedule the **TD-WEXTL** background process. This process generates a To Do entry for every record in the workflow event exception table.

After correcting the cause of the error, drill into the [Workflow Process - Events](#) tab and change the status from **Error** to **Pending**. The next time the event activation process runs, it attempts to process this record again.

## Uploading Notifications

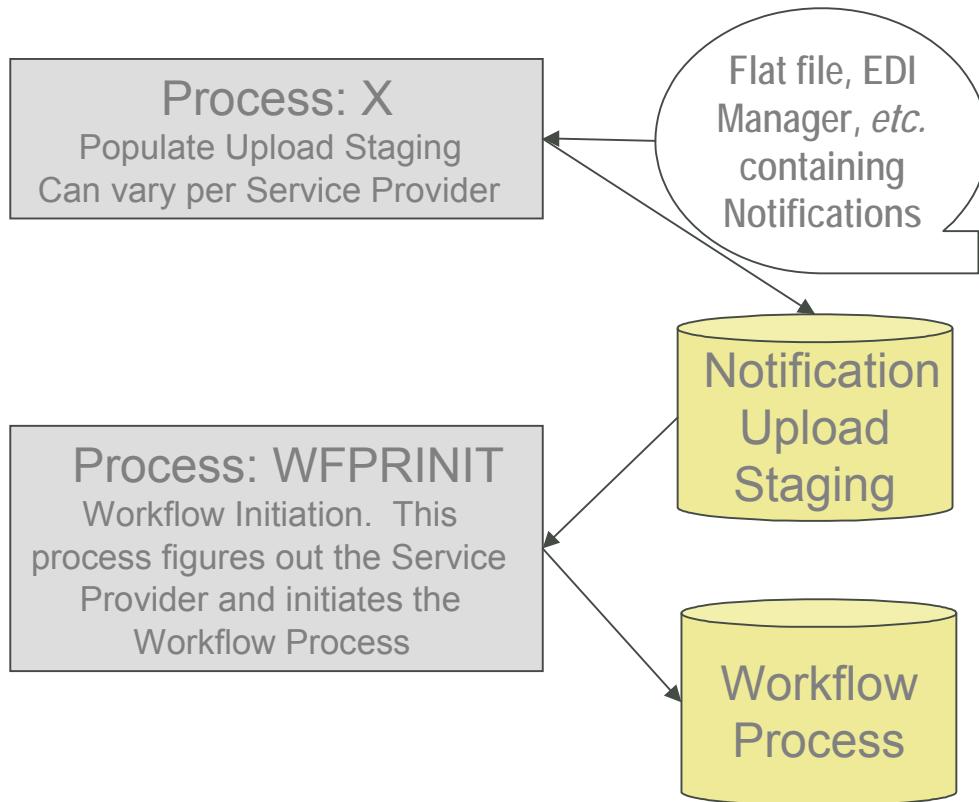
This section describes how the system uploads notifications. Refer to [Uploading Notifications Into The System](#) for more information.

### Contents

- [Notification Upload Background Process](#)
- [Notification Upload Staging Maintenance](#)
- [Notification Upload Staging - Sample Upload Extension Record](#)
- [Notification Upload Exception](#)

## Notification Upload Background Process

The following diagram illustrates the processes involved in the uploading of notifications into the system.



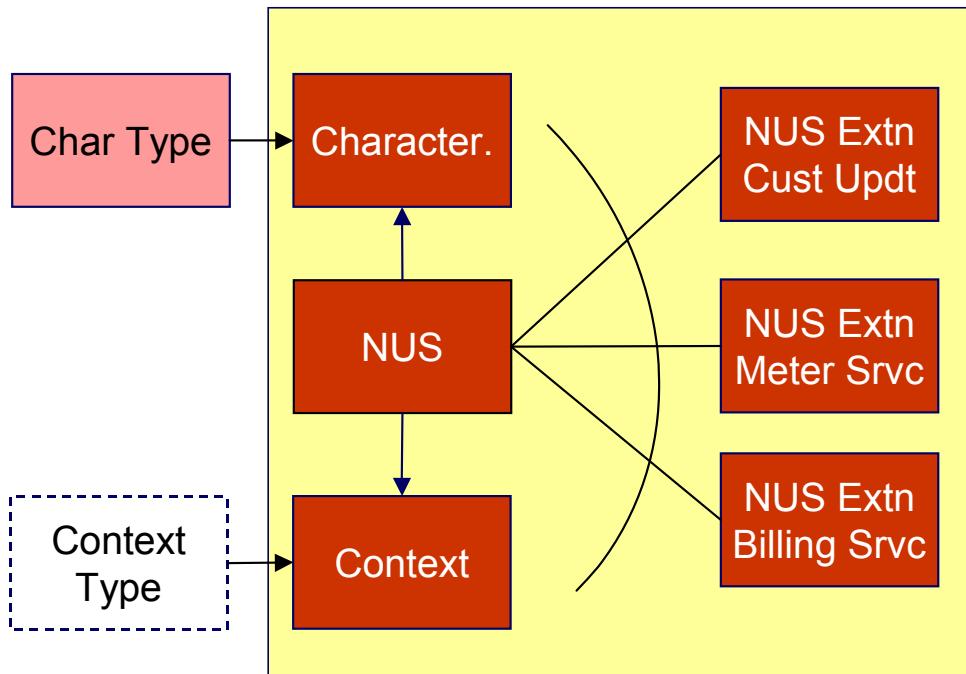
The topics in this section describe how these processes work.

### Contents

- [Process X - Populate Notification Upload Staging](#)
- [Process WFPRINIT - Initiate Workflow Processes To Execute Notification Upload Staging R Validation of Data Related to the Notification Upload Staging Records](#)

### **Process X - Populate Notification Upload Staging**

Process X refers to the mechanism used by your organization to populate the various staging tables (shown in the yellow section of the following entity relationship diagram).



The NUS record contains only key header information, but it does not contain detailed information that would be required to process the notification. To define the other required information, like Account ID and meter number, you have three options:

- **NUS Characteristics:** This is a collection of characteristic values, where the characteristic type describes the data that is contained in the characteristic values. In order to set this up correctly, the appropriate [characteristic types](#) need to be defined.
- **NUS Context:** This is a collection of context values, where the context type describes the data that is contained in the context values. In order to set this up correctly, the appropriate context types need to be defined by your implementers.
- **NUS Extension:** This is a separate table, with explicit columns defining the data. In order to set this up correctly, a new table and new user interface will need to be created by your implementers, along with a new Extension type.

For any of the methods, the first Workflow Event in your workflow process will have the responsibility to retrieve the core information from the Characteristic collection, the Context collection or the Extension record and populate the workflow process context or characteristic collection accordingly.

The topics in this section describe each of these tables.

## Contents

- [Notification Upload Staging](#)
- [Notification Upload Staging Context](#)
- [Notification Upload Staging Characteristics](#)
- [Notification Upload Staging Extension - Base Sample](#)

### Notification Upload Staging

You must create an upload staging record for each notification being processed in the system. The name of this table is [CI\\_NT\\_UP](#). The following table describes each column on this table.

Column Name	Length	Req'd	Data Type	Comments
NT_UP_ID	12	Y	A/N	<p>This is a sequential unique key. This key is referenced on the NUS extension record. (The NUS extension record contains almost all of the fields associated with the notification.)</p> <p>This key is a sequential number so you can use a database function to assign the value when populating the table.</p>
NT_XID_CD	30	Y	A/N	<p>This identifies the external system that sent the notification (e.g., a DUNS number).</p> <p>This must correspond with the prime key of one of your external systems. Refer to <a href="#">Designing External Systems</a> for more information.</p>
NT_UP_XTYPE_CD	30	Y	A/N	<p>This identifies the transaction type of the notification.</p> <p>This must correspond with the prime key of one of your notification upload types. Refer to <a href="#">Designing Notification Upload Types</a> for more information.</p>
NT_UP_STAT_FLG	2	Y	A/N	This must be set to <i>P</i> for <i>Pending</i> .
NT_UP_EXTSN_FLG	4	Y	A/N	<p>This identifies the type of notification upload staging extension record that contains the details of this notification upload. Valid values: <i>NONE</i>, <i>SPL</i>. Use <i>NONE</i> if you will use the context collection for your related data.</p>
CRE_DTTM	26	Y	DateTime	This is the date / time on which the notification upload staging record was created. This must be populated with the current date / time.

### Notification Upload Staging Context

If desired, you may insert rows into the upload staging context collection. The context collection enables you to define all the core information associated with the upload staging record. Context flag values will need to be defined for every piece of data you wish to capture. This option may be used in favor of the extension record option, defined below, because you do not need to define new database tables or new user interfaces for each new notification upload type.

The name of this table is [CI\\_NT\\_UP\\_CTXT](#). The following table describes each column on this table.

Column Name	Length	Req'd	Data Type	Comments
NT_UP_ID	12	Y	A/N	This is the unique key of the NUS record to which this collection is linked.

NT_UP_CTXT_FLG	4	Y	A	This field describes the data that is stored in the Context Value. Refer to <a href="#">Notification Upload Staging</a> for more information.
CTXT_VAL	50	Y	A	This is the value of the context.

### Notification Upload Staging Characteristics

If desired, you may insert rows into the upload staging characteristics collection. The characteristics collection enables you to define all the core information associated with the upload staging record. [Characteristic types](#) will need to be defined for every piece of data you wish to capture. This option may be used in favor of the extension record option, defined below, because you do not need to define new database tables or new user interfaces for each new notification upload type.

The name of this table is [CI\\_NT\\_UP\\_CHAR](#). The following table describes each column on this table.

Column Name	Length	Req'd	Data Type	Comments
NT_UP_ID	12	Y	A/N	This is the unique key of the NUS record to which this collection is linked.
CHAR_TYPE_CD	8	Y	A	This field describes the data that is stored in the characteristic value field. Refer to <a href="#">characteristic type</a> for more information.
SEQ_NUM	3	Y	N	This value is used to differentiate multiple characteristics of the same type. Values are typically incremented by 10.
CHAR_VAL	16	N	A	Populate this field if your <a href="#">characteristic type</a> is <i>predefined</i> .
ADHOC_CHAR_VAL	254	N	A	Populate this field if your <a href="#">characteristic type</a> is <i>ad-hoc</i> or <i>file location</i> .
CHAR_VAL_FK1 – CHAR_VAL_FK5	50 each	N	A	Populate these fields if your <a href="#">characteristic type</a> is <i>foreign key reference</i> . Up to five columns of 50 bytes each are provided to accommodate compound keys.

### Notification Upload Staging Extension - Base Sample

A notification upload staging EXTENSION record may be used in lieu of Context to contain the core information associated with a notification upload record. This option enables you to be explicit in your definition of the fields that must be captured for each notification upload type.

The sample extension record that is supplied with the system is just that - a sample. If you use this option, you will need one such extension record for each different type of notification upload record layout you support. It's important to understand that all of your different notification upload extension record types must reference the [core](#) [Notification Upload Staging](#) record because the system uses the [core](#) record to drive the notification upload process.

The name of the [sample](#) notification upload extension record is [CI\\_NT\\_UP\\_XSPL](#). The following table describes each column on this table.

Column Name	Length	Req'd	Data	Comments
-------------	--------	-------	------	----------

			Type	
NT_UP_ID	12	Y	A/N	This is the reference to the <a href="#">Notification Upload Staging</a> record that this record is an extension of.
REC_DTTM	26	Y	Date/Time	This is the date / time the record was received.
RESPONSE_CODE	1	Y	A/N	This column is used when the notification is an indication of the acceptance or a rejection of an earlier notification request. If specified, this must be either <b>A</b> - accept or <b>R</b> - reject.
SDR_ACCT_ID	20	Y	A/N	This is the customer's account ID in the sender's system.
RCV_ACCT_ID	20	Y	A/N	This is the customer's account ID in the receiver's system.
SDR_REC_ID	16	Y	A/N	This is a transaction identifier supplied so responses to this record may be identified. The contents of this field will be returned on notification download records that respond to this notification upload record.
RCV_REC_ID	16	Y	A/N	This is the transaction identifier received in a previous request record. This field is only populated if the notification being uploaded is an acceptance / rejection of a previously sent notification download record.
BILL_OPT_FLG	4	Y	A/N	This defines the billing relationship between the service provider and your organization. This value is used in conjunction with BILL_CALC_OPT_FLG to determine the service provider associated with the NT_XID_CD on the CI_NT_UP record. Valid values are: <b>DUAL</b> , <b>ESP</b> , <b>UDC</b> .
BILL_CALC_OPT_FLG	4	Y	A/N	This defines the bill calculation method used when your organization provides billing services for the service provider. This value is used in conjunction with BILL_CALC_FLG to determine the service provider associated with the NT_XID_CD on the CI_NT_UP record. Valid values are: <b>BILL</b> (bill ready), <b>RATE</b> (rate ready).
MTR_NBR	16	Y	A/N	The customer's meter number. This is used to help identify the service agreement associated with the notification.
BILLING_DATE	2	Y	N	This is the day of the month used to derive the service cycle define on SA relationship records associated with service providers

				who provide MR services.
POSTAL_CD	12	Y	A/N	The service address's postal code. This is used as a crosscheck to ensure the notification is referencing the correct customer.

## Process WFPRINIT - Initiate Workflow Processes To Execute Notification Upload Staging Records

This process creates a workflow process for all [Notification Upload Staging](#) records in the **Pending** state. It determines the workflow process using a combination of the External Id and the NUS Type. Refer to [What Type Of Workflow Process Is Created](#) for more information. It stamps the NUS ID onto the newly created workflow process.

If the External ID on the staging record is not a valid External ID in our system, rather than creating an error, this process will find a workflow process template with a notification condition of **Unknown Notification ID** and create a workflow process. If the NUS Type on the staging record is not a valid NUS Type in our system, rather than creating an error, this process will find a workflow process template with a notification condition of **Unknown Notification ID** and create a workflow process. Refer to [Setting Up Workflow Process Templates](#) for more information. For these cases, the status of the notification upload staging record will be set to **Complete**.

Any errors detected during this process will be written to the [Notification Upload Exception](#) table. You can fix errors using the [Notification Upload Staging](#) page (don't forget to change the record's status back to **Pending**).

Refer to [Uploading Notifications Into The System](#) for more information.

## Validation of Data Related to the Notification Upload Staging Records

As mentioned above, the sole purpose of the Initiate Workflow Process (WFPRINIT) is to successfully create a workflow process. Once a workflow process is created, the NUS status is set to **Complete**.

At this point, neither the extension record, the context collection nor the characteristics collection have been processed. It is the responsibility of the first event in the workflow process to validate the fields in the extension record, the context collection and/or the characteristics collection. This algorithm will decide what to do when a problem is detected with the data. Some possible options are

- Set the algorithm status to **Fail**. Use this option when a critical error is detected, which probably cannot be fixed by a user. This will cause the remaining events to be canceled.
- Set the algorithm status to **Error**. Use this option when an error is detected, which can be corrected by a user. This will cause an [exception](#) record to be created. The user should be able to correct the source data in the extension record, the context collection or the characteristics collection, change the status back to **Pending** and rerun the Workflow Event Activation process. Note that if the validation algorithm uses this option, the ability to rerun this event should be handled by the coding of this algorithm. It should prevent duplicate logic from being performed.

- Send a To Do list entry and continue the workflow process by setting the algorithm status to **Complete**. Use this option when someone should be alerted of an unusual condition, but the process should not be halted. For this option, the To Do Type and User to use for posting the To Do entry should be parameters to the algorithm.

**Note.** We said that a possible correction for the validation workflow event in the **Error** is to modify data in the extension record, the context collection or the characteristics collection. You may find this a bit unusual because the parent NUS record is in the status of **Complete**. Remember that this NUS status indicates whether or not the system was able to successfully create a workflow process. There is no status for the data related to the NUS – only the status of the event, which validates it.

## Notification Upload Staging Maintenance

The Notification Upload Staging page is used to view notification upload records.

Refer to [Process X - Populate Notification Upload Staging](#) for more information about these records.

### Contents

[Notification Upload Staging - Main](#)  
[Notification Upload Staging - Characteristics/Context](#)

### Notification Upload Staging - Main

Use this page to define common attributes of notification upload staging records. Open this page using **Workflow and Notification, Notification Upload Staging, Main**.

#### Description of Page

The **Upload Staging ID** is the unique identifier of the notification upload staging (NUS) record.

**Status** shows the state of the NUS record. Potential values are: **Pending**, **Complete**, **Error**. To have the system reprocess a record that's in **Error**, change the status back to **Pending**. The system will revalidate the record the next time the notification upload process executes.

**Create Date and Time** are just that.

**External System** identifies the sender of the notification.

**Notification Upload Type** identifies the transaction type of the notification.

**Extension** identifies the type of notification upload staging extension record that contains the details of this notification upload. Valid values: **No Extension**, **Sample Extension Record**. You would use **No Extension** when you plan to define the data related to the NUS record in the characteristic collection or in the context collection rather than in an extension table. If your [notification upload type](#) indicates a navigation option associated with your extension record, a drill down button is enabled adjacent to this field.

**Default Note.** The extension value defaults from the value specified on the notification upload type.

**Note.** The values for this field are customizable using the Lookup table. This field name is NT\_UP\_EXTSN\_FLG.

If a **Workflow Process** has been created for this NUS record, information about the workflow process will be displayed.

This bottom of this page contains a [tree](#) that shows the various objects linked to the notification upload staging record along with information about the workflow process, if one has been created for the NUS. You can use this tree to both view high-level information about these objects and to transfer to the respective page in which an object is maintained.

### Notification Upload Staging - Characteristics/Context

Use this page to either characteristic records or context records to define data related to the NUS record. Either collection may be used as an alternative to defining an extension record. Refer to [Process X - Populate Notification Upload Staging](#) for more information.

Open this page using **Workflow and Notification, Notification Upload Staging, Characteristics/Context**.

#### Description of Page

The **Upload Staging ID** is the unique identifier of the notification upload staging (NUS) record.

The **Characteristics** collection enables you to define a collection of **Characteristic Types** and **Characteristic Values** defining data related to the NUS record.

The **Context** collection enables you to define a collection of **Context Types** and **Context Values** defining data related to the NUS record.

**Note.** The values for this field are customizable using the Lookup table. This field name is NT\_UP\_CTXT\_FLG.

**Note.** You will notice that both the characteristic collection data and the context collection data may be modified even when the parent NUS record is **Complete**. This is to enable a user to correct problems with this data as detected by the validation event. Refer to [Validation of Data Related to the Notification Upload Staging Records](#) for more information.

## Notification Upload Staging - Sample Upload Extension Record

The Notification Upload Staging page is used to maintain notification upload records.

Refer to [Process X - Populate Notification Upload Staging](#) for more information about these records.

Open this page using **Workflow and Notification, Sample Upload Extension**.

#### Description of Page

**Receive Date and Time** are just that.

**Response Code** is used when the notification is an indication of the acceptance or a rejection of an earlier notification request. If specified, this must be either **Accept** or **Reject**.

**Sender's Customer's Account ID** is the customer's account ID in the sender's system.

**Receiver's Customer's Acct ID** is the customer's account ID in your system.

**Sender Record ID** is a transaction identifier supplied so responses to this record may be identified. The contents of this field will be returned on notification download records that respond to this notification upload record.

**Receiver Record ID** is the transaction identifier received in a previous request record. This field is only populated if the notification being uploaded is an acceptance / rejection of a previously sent notification download record.

**Billing Option** defines the billing relationship between the service provider and your organization. This value is used in conjunction with **Billing Calculation Option** to determine the service provider associated with the sender's external system (stored on the related notification upload staging record). Valid values are: **Dual Billing**, **ESP Consolidated**, **UDC Consolidated**.

**Billing Calculation Option** defines the bill calculation method used when your organization provides billing services for the service provider. This value is used in conjunction with **Billing Option** to determine the service provider associated with the sender's external system (stored on the related notification upload staging record). Valid values are: **Bill Ready**, **Rate Ready**.

**Meter Number** is the customer's meter number. This is used to help identify the service agreement associated with the notification.

**Billing Date** is the day of the month used to derive the service cycle define on SA relationship records associated with service providers who provide meter reading services.

**Postal Code** is the service address's postal code. This is used as a cross check to ensure the notification is referencing the correct customer.

**Note.** You will notice that the extension data may be modified even when the parent NUS record is Complete. This is to enable a user to correct problems with this data as detected by the validation event. Refer to [Validation of Data Related to the Notification Upload Staging Records](#) for more information.

## Notification Upload Exception

A record is written the notification upload exception record for every notification upload staging record that is in error.

To view the messages associated with the exception records, schedule the [TD-NTUPL](#) background process. This process generates a To Do entry for every record in the notification upload exception table.

After correcting the cause of the error, drill into the [Notification Upload Staging](#) page and change the status from **Error** to **Pending**. The next time the notification upload process runs, it attempts to process this record again.

# Downloading Notifications

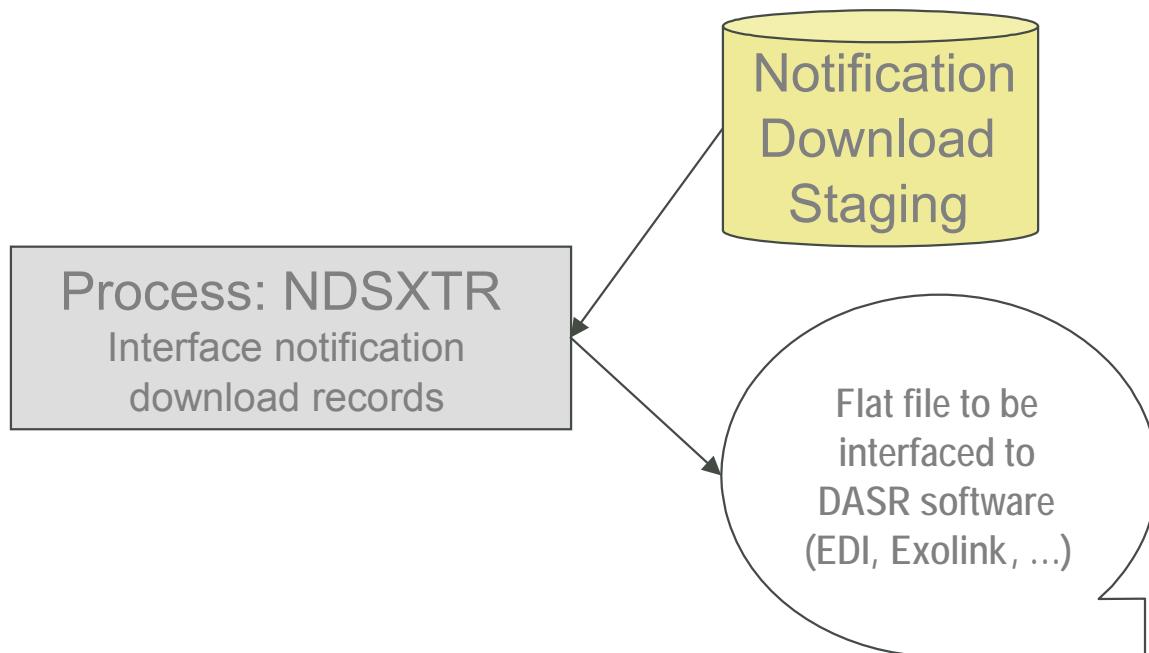
This section describes how the system downloads notifications. Refer to [How Are Notifications Sent Out Of The System?](#) for more information. Information may be extracted from the system using a batch process, which runs in the background or using the XAI tool, which continuously searches for new NDS records and builds XML documents to send to an external recipient.

## Contents

- [Notification Download Background Processes](#)
- [Notification and XAI](#)
- [NDS Contains Interesting Data](#)
- [Notification Download Staging](#)
- [Notification Download Exception](#)

## Notification Download Background Processes

The following diagram illustrates how a batch process, which runs in the background may be used to route notifications out of the system.



The topics in this section describe how this process works.

### Process NDSXTR - Interface Notification Download Records (out of the system)

This process reads all notification download staging records marked with a given batch control ID & run number and creates the flat file that's passed to the outgoing notification software. Every NDS record references the recipient service provider and a notification download type. This process uses the service provider's notification download profile to determine the format of records interfaced out of the system. Refer to [Designing Notification Downloads](#) for more information.

**This process is rerunnable.** You can reproduce the flat file at any time. Simply request this job and specify the run number associated with the historic run.

If you require different routing methods, you will need to create additional versions of this program.

## Notification and XAI

NDS records may also be processed using the XAI tool.

Refer to [Notification and XAI](#) for more information.

## NDS Contains Interesting Data

The NDS record also contains interesting data

- The main use for this is to indicate the workflow process and event that created the NDS
- The process that extracts the data may then find the other data related to this NDS by looking at the workflow process context or characteristics

This interesting data may be stored in one of two ways

- Through NDS context collection. Use this method if your NDS is interfaced to the outside world using XAI. Refer to [Designing Notification Download Types](#) for more information.
- Through NDS characteristics collection. For NDS records that are interfaced through a batch process, it is expected that using characteristics is the preferred method because if your characteristic references a foreign key, your user can easily drill down to the related record.

**Note.** The algorithms provided with the base product use the context collection to populate the interesting data for an NDS.

## Notification Download Staging

The Notification Download Staging page is used to view notification download records. This is especially useful if you send out a notification and request a response.

Refer to [Notification Download Background Processes](#) for more information about these records.

### Contents

- [Notification Download Staging - Main](#)
- [Notification Download Staging - Characteristics/Context](#)
- [Notification Download Staging - XAI Download Staging](#)

### Notification Download Staging - Main

Open this page using **Workflow and Notification, Notification Download Staging**.

### Description of Page

The **Download Staging ID** is the unique identifier of the notification download staging (NDS) record.

**Service Provider** is the service provider to whom this download staging record will be sent. The description of the service provider is displayed adjacent.

**Notification Download Type** indicates the reason for this notification. The description is displayed adjacent.

Refer to [Designing Notification Download Types](#) for more information about notification download types.

The **Processing Method** indicates whether this record will be processed by a **Batch** extract process or through the **XAI** tool. The processing method defined on the Notification Download Profile will be used to populate this value.

If the processing method is **Batch**, then **NDS Extract Process** indicates the process that will extract this record. This value is populated based on the service provider's notification download profile. **NDS Extract Run** indicates in which batch run this record was extracted or will be extracted.

Refer to [Designing Notification Download Profiles](#) for more information about the NDS Extract Process.

The **Retry Count** is used by the XAI tool to keep track of how many times the tool tried to process this record and could not process the record, resulting in an error.

The **Create Date/Time** indicates the date and time that this record was created.

**Status** defines the state of the NDS record. This is only used when the processing method is **XAI** and will be protected when the processing method is **Batch**. Refer to [Lifecycle of Notification Download Staging](#) for more information.

This bottom of this page contains a [tree](#) that shows the various objects linked to the notification download staging record. You can use this tree to both view high-level information about these objects and to transfer to the respective page in which an object is maintained.

## Notification Download Staging - Characteristics/Context

This page displays the characteristics and/or context associated with the download staging record. Refer to [NDS contains interesting data](#) for more information.

Open this page using **Workflow and Notification, Notification Download Staging, Characteristics/Context**.

### Description of Page

The **Download Staging ID** is the unique identifier of the notification download staging (NDS) record.

If **Characteristics** are used by the algorithm that created this record, to record information related to the record, this grid displays the **Characteristic Type** and **Characteristic Value** for each entity associated with the download staging record.

If **Context** entries are used by the algorithm that created this record, to record information related to the record, this grid displays the **Context Type** and **Context Value** for each entity associated with the download staging record.

**Note.** The values for **Context Type** are customizable using the Lookup table. This field name is NT\_DWN\_CTXT\_FLG.

## Notification Download Staging - XAI Download Staging

Open this page using **Workflow and Notification**, **Notification Download Staging**, **XAI Download Staging**.

### Description of Page

For XAI messages, an NDS Type may point to multiple XAI Route Types. Each route type is linked to an XAI sender. The system keeps track of each routing and each response. This collection of **XAI Route Type** includes the **XAI Download Status** and **Message**. To look at more detail, drill down into a specific XAI Routing and you are transferred to the [XAI Download Staging](#) page, where you may view the XML Request and XML Response.

## Notification Download Exception

A record is written the notification download exception record for every notification download staging record that is in error. This is only applicable to records with a processing method of **XAI**.

To view the messages associated with the exception records, schedule the **TD-NTDWN** background process. This process generates a To Do entry for every record in the notification download exception table.

Refer to [Lifecycle of Notification Download Staging](#) for more information.

# Overdue Financial Obligations

The system periodically monitors how much your customers owe to ensure they haven't violated your overdue rules. When a violation is detected, the system creates an overdue process. The overdue process contains the events meant to prod the customer to pay (e.g., letters, disconnect field activities, To Do entries, write-off outstanding debt, etc.). This section describes how to manage your overdue (i.e., collections) processing.

**Collecting on unpaid bills.** 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 the 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 [Credit and Collections](#).

## Contents

- [Overdue Processing Background Information](#)
- [Overdue Process Maintenance](#)
- [Cut Process Maintenance](#)
- [Payment Arrangement for Bills](#)

## Overdue Processing Background Information

In the section, [The Big Picture Of Overdue Processing](#), we describe how overdue processing works and how to set up the control tables that automate most functions. You will find that many of your questions regarding when and how overdue and cut processes are created and canceled are described in this section.

## Overdue Process Maintenance

An overdue process is a series of events (e.g., letters, To Do entries) meant to encourage an account to pay its delinquent debt. The topics in this section describe the pages on which overdue processes are maintained.

Refer to [The Big Picture Of Overdue Processes](#) for background information.

## Contents

- [Overdue Process - Main](#)
- [Overdue Process - Events](#)
- [Overdue Process - Log](#)
- [How To Perform Common Overdue Process Functions](#)

## Overdue Process - Main

The Main page contains core overdue process information. Open this page using **Credit & Collections, Overdue Process** from the main menu.

### Description of Page

Refer to [How To Perform Common Overdue Process Functions](#) for more instructions describing how to use this page.

**Overdue Process** contains a concatenation of important information about the process. **Overdue Process ID** is the system-assigned unique identifier of the process. These values only appear after the overdue process is added to the database.

**Summary information may be overridden.** Refer to [Overdue Process Information Is Overridable](#) for how your implementation can override the summary information that appears throughout the system.

**Account ID** identifies the overdue process's account.

**Multiple overdue processes may be linked to an account.** It's important to be aware that it's possible for multiple, active overdue processes to be linked to an account. The [Alert zone](#) will contain a summary of the account's overdue processes.

**Status** defines the state of the overdue process. The following values may exist:

- **Active, Events Pending.** Overdue processes are initially created in this state. An overdue process remains in this state until there are no **Pending** or **Waiting** events.
- **Inactive, Canceled by User.** An overdue process will exist in this state when it's been manually canceled by a user. Navigate to the **Log** tab to see when this happened and who did it.
- **Inactive, Canceled by System.** An overdue process will exist in this state when it's been canceled by the system (typically because the overdue obligations were satisfied). Navigate to the **Log** tab to see when this happened.
- **Inactive, Completed.** An overdue process will exist in this state when the system completes its last event. Navigate to the **Log** tab to see when this happened.

The **Cancel** button appears if the overdue process is **Active, Events Pending** and the process's **Overdue Process Template** has a **Cancel Logic** plug-in. Click this button to [cancel the process](#).

The **Trigger Events** button appears if the overdue process is **Active, Events Pending**. Click this button to activate all pending events that are [ready for activation](#).

**If you can't wait for the Overdue / Cut Event Manager to run.** The [Overdue / Cut Event Manager](#) is a background process that activates events on their trigger date. If you don't want to wait for this process to run, you can click the **Trigger Events** button on the **Main** tab to activate the process's events (that are ready for activation).

**Overdue Process Template** defines the template that was used to [create the overdue process's events](#). You can override these events on the **Events** page. This field is unprotected when the process is **Active** and all events are in either the **Pending** or **Canceled** states.

**Changing the template.** If you change the template, the system will delete the events and replace them with the new template's events.

**Start Date/Time** defines the start date/time of the overdue process. This field is protected after the overdue process is saved on the database. This field is used to derive the trigger date of some overdue events when the process is first created. Subsequent changes to the start date change the trigger dates on the events accordingly.

**Inactive Date/Time** is the date and time that the overdue process became **Inactive**. This field is hidden if the process is **Active**. You can see more details about when and how the inactivation occurred by navigating to the **Log** tab.

**Base Date** is reserved for future releases. This field is protected when the overdue process is **Inactive**.

Enter any **Comments** about the overdue process. This field is protected when the overdue process is **Inactive**.

The **Collecting On** grid holds the [overdue objects](#). For example, if this process manages overdue bills, the grid contains a list of the bills being collected on by this process. This grid is unprotected when the process is **Active**. The following points describe the columns that appear in the grid:

- Press the + button to add a new overdue object to the process.
- Press the - button to remove an overdue object to the process.
- The next column contains the unique identifier of the overdue object. For example, if this process manages overdue bills, this column will contain the bill ID. The type of object that's specified in this column is control by the [overdue process template](#).
- **Original Amount** contains the [original amount](#) of the overdue object.
- **Unpaid Amount** contains the [unpaid amount](#) of the overdue object.

The [Trees](#) at the bottom of the page shows a variety of information about the overdue process including its events and cut processes. You can click on hyperlinked tree nodes to navigate to the page on which the related object is maintained.

## Overdue Process - Events

This page contains the activities that are performed to persuade the customer to pay the outstanding debt. Open this page using **Credit & Collections**, **Overdue Process** and then navigate to the **Events** tab.

### Description of Page

Refer to [How To Perform Common Overdue Process Functions](#) for more instructions describing how to use this page.

Refer to the first tab for a description of **Overdue Process** and **Overdue Process ID**.

The **Overdue Events** scroll contains the process's overdue events. Refer to [Overdue Processes Are Created From Templates](#) for information about how the system defaults a process's events from its template.

- Note, all information in the scroll is protected if the event's Status is **Complete** or **Canceled**.
- **Event Sequence** is the unique identifier of the event.
- **Status** defines the [state of the event](#).
- The **Cancel** button appears if the event is **Pending** or **Waiting**, and the event's **Overdue Event Type** has a **Cancel Logic** plug-in. Click this button to cancel the event.
- The **Overdue Event Type** defines the event's activity (e.g., Email sent, a To Do entry is generated, a letter is sent).
- The remaining fields work in unison:
  - If **Dependent on Other Events** is turned on, the event can only be triggered after the events specified in the **Event Dependencies** grid are all **Complete** or **Canceled**. You use the **Days After** field to define when this event is activated. For example, if you enter 5, this event will be activated 5 days after all of the dependent events are completed / cancelled. If you enter 0, this event will be activated immediately after the dependent events are completed / cancelled. Refer to [Calendar vs Work Days](#) for a descriptions of how days are counted.  
If all dependent events are already **Completed** or **Canceled**, you cannot specify **Days After**. Rather, you must enter the desired activation date in the **Trigger Date** field.
  - If **Dependent on Other Events** is turned off, enter the desired activation date in the **Trigger Date** field. If this field is off, the **Days After** field and the **Event Dependencies** grid are hidden.

If you can't wait for the Overdue / Cut Event Manager to run. The [Overdue / Cut Event Manager](#) is a background process that activates events on their trigger date. If you don't want to wait for this process to run, you can click the **Trigger Events** button to activate the process's events (that are ready for activation).

## Overdue Process - Log

This page contains [log entries](#) that highlight significant events in the process's life. Open this page using **Credit & Collections, Overdue Process** and then navigate to the **Events** tab.

### Description of Page

Please note the following about the entries in this grid:

- The system automatically creates log entries when significant events occur. Refer to [Overdue Log](#) for more information.
- You can manually add a log entry to an Active process by pressing the + button and enter the **Details**. You cannot modify or delete this information after saving it.
- The following information appears in the grid:
  - **Date/Time** contains the date and time the log entry was created.

- **Details** contain the user-specified or system-generated information about the log entry.
- **Relate Object** is populated on log entries that were created to record the creation of some other object. For example, if an overdue event creates a customer contact, the related object contains information about the customer contact. Please note that if the object description is shown in blue, you can click on the object's description to drill down to the object.
- **Related Process / Event** appears if the log entry was created by an overdue event, a cut process or a cut event. It references the unique identifier of the process / event.
- **Log User** contains the user who caused the log entry to be created.
- **Log Type** indicates how the log entry was created. The possible values are:
  - **User.** A user manually created this entry.
  - **System.** This system created this entry.

## How To Perform Common Overdue Process Functions

The topics in this section describe how to perform common overdue process maintenance functions. Refer to [The Big Picture Of Overdue Processing](#) for high-level information about overdue processing.

### Contents

- [How To Create An Overdue Process](#)
- [How To Change Overdue Events](#)
- [How To Cancel An Overdue Process](#)

### How To Create An Overdue Process

99.9% of all overdue processes are created by the [Overdue Monitor](#) and require no human intervention. The other 0.1% are created by users on-line / real time. The following points describe how to create the 0.1%.

- Use **Control Central** to choose the account that requires a new overdue process.
- After the account is populated on Control Central, choose the **Overdue Process +** option on the account context menu to transfer to the overdue process transaction in add mode for the account.
- After the [Overdue Process - Main](#) page appears, specify the appropriate **Overdue Process Template**. The template is used to default the process's events (the template also controls numerous business rules, for example, when it will be canceled, how it will be canceled, etc). Refer to [How To Change Overdue Events](#) for a description of how you can override these events.
- Enter the overdue object(s) in the **Collecting On** grid. You must define at least one object.
- Save the overdue process.

### How To Change Overdue Events

When an overdue process is first created, it has one or more overdue events. The events are the activities that will be performed to persuade the customer to pay the outstanding debt.

The number and type of events that are created when an overdue process is initiated are defined on the overdue process's overdue process template. The following points describe how to add / change / delete events on an overdue process if the defaulted events are not satisfactory.

- Use **Control Central** to choose the account with the overdue process whose events need to be changed.
- After the account is populated on Control Central, choose the **Overdue Process** option on the account context menu to transfer to the overdue process transaction in update mode for the account. Note, an account's **Active** overdue processes can be selected from the [Alerts zone](#).
- To add a new event, transfer to the [Overdue Process - Events](#) tab and press the + button in the **Overdue Events** scroll. At this point, the event has not been added to the database; rather, it just exists in memory. Before you add the event to the database, you must specify the following information:
  - Choose an **Event Sequence** so that the new event will be positioned properly in respect of the other events.
  - Choose a **Status** of **Pending**.
  - Choose the desired **Overdue Event Type**.
  - If the activation of the new event is dependent on the successful completion of earlier events, turn on **Dep on Other Event** and then:
    - specify the sequences of the dependent events in **Event Dependencies** and
    - specify the how many days after the completion of the last dependent event that the new event should be triggered.
  - If the activation of the new event is NOT dependent on the successful completion of earlier events, turn off **Dep on Other Event** and use **Trigger Date** to define the date on which the event should be activated (i.e., completed).
- To delete an existing event, click on the event in the tree on [Overdue Process - Main](#). This will transfer you to the **Events** tab where you can click the - button to remove the event. At this point, the event has not been removed from the database; rather, it's been removed in memory.
- To change an existing event, click on the event in the tree on [Overdue Process - Main](#). This will transfer you to the **Events** tab where you can make the desired changes.
- After all desired changes have been made, save the overdue process.

## How To Cancel An Overdue Process

The system will cancel an overdue process when the [cancel criteria defined on the overdue process are satisfied](#). The following points describe how to manually cancel an overdue process.

- Use **Control Central** to choose the account with the overdue process to be cancelled.
- After the account is populated on Control Central, choose the **Overdue Process** option on the account context menu to transfer to the overdue process transaction in update mode for the account. A list of all overdue processes associated with the account appears. If only one overdue process exists, it is automatically selected for you. Note, an account's **Active** overdue processes can be selected from the [Alerts zone](#).
- On the **Main** tab, click **Cancel** and answer any prompts.

## Cut Process Maintenance

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. The topics in this section describe the pages on which cut processes are maintained.

Refer to [The Big Picture Of Cut Processes](#) for background information.

### Contents

- [Cut Process - Main](#)
- [Cut Process - Events](#)
- [How To Perform Common Cut Process Functions](#)

## Cut Process - Main

The Main page contains core cut process information. Open this page using **Credit & Collections, Cut Process** from the main menu.

### Description of Page

Refer to [How To Perform Common Cut Process Functions](#) for more instructions describing how to use this page.

**Cut Process** contains a concatenation of important information about the process. **Cut Process ID** is the system-assigned unique identifier of the process. These values only appear after the cut process is added to the database.

**Summary information may be overridden.** Refer to [Cut Process Information Is Overridable](#) for how your implementation can override the summary information that appears throughout the system.

**Overdue Process** and **Overdue Event** define overdue event that created the process. Refer to [Cut Processes Exist Under An Overdue Process](#) for more information.

**Service Agreement** identifies the service agreement being cut.

**Premise** identifies the service agreement's characteristic premise (if any).

**Multiple cut processes may be linked to an overdue process / event.** It's important to be aware that it's possible for multiple, active cut processes to be linked to the overdue event. You can see all cut processes associated with an overdue process on the tree of [Overdue Process - Main](#). The [Alert zone](#) will contain a summary of the account's cut processes.

**Status** defines the state of the cut process. The following values may exist:

- **Active, Events Pending.** Cut processes are initially created in this state. A cut process remains in this state until there are no **Pending** or **Waiting** events.

- **Inactive, Canceled by User.** A cut process will exist in this state when it's been manually canceled by a user. Navigate to the overdue process's [Log](#) tab to see when this happened and who did it.
- **Inactive, Canceled by System.** A cut process will exist in this state when it's been canceled by the system (typically because the overdue obligations on the overdue process were satisfied). Navigate to the [Log](#) tab to see when this happened.
- **Inactive, Completed.** A cut process will exist in this state when the system completes its last event. Navigate to the overdue process's [Log](#) tab to see when this happened.

The **Cancel** button appears if the cut process is **Active, Events Pending** and the process's **Cut Process Template** has a **Cancel Logic** plug-in. Click this button to [cancel the process](#).

The **Trigger Events** button appears if the cut process is **Active, Events Pending**. Click this button to activate all pending events that are [ready for activation](#).

**If you can't wait for the Overdue / Cut Event Manager to run.** The [Overdue / Cut Event Manager](#) is a background process that activates events on their trigger date. If you don't want to wait for this process to run, you can click the **Trigger Events** button to activate the process's events (that are ready for activation).

**Cut Process Template** defines the template that was used to [create the cut process's events](#). You can override these events on the **Events** page. This field is unprotected when the process is **Active** and all events are in either the **Pending** or **Canceled** states.

**Changing the template.** If you change the template, the system will delete the events and replace them with the new template's events.

**Start Date/Time** defines the start date/time of the cut process. This field is protected after the cut process is saved on the database. This field is used to derive the trigger date of some cut events when the process is first created. If you need to subsequently change the trigger date on the events, navigate to the **Events** tab and change the dates manually. Alternatively, you can perform the following "trick":

- Change the **Cut Process Template** to something (anything). Doing this causes the **Start Date/Time** to become unprotected.
- Specify the base date of the cut events in the newly unprotected **Start Date/Time**.
- Change the **Cut Process Template** back to the desired value. When you change the template, the system removes the events and creates new ones using the **Start Date/Time**.

**Inactive Date/Time** is the date and time that the cut process became **Inactive**. This field is hidden if the process is **Active**. You can see more details about when and how the inactivation occurred by navigating to the overdue process's [Log](#) tab.

Enter any **Comments** about the cut process. This field is protected when the cut process is **Inactive**.

The [tree](#) at the bottom of the page shows a variety of information about the cut process including its events. You can click on hyperlinked tree nodes to navigate to the page on which the related object is maintained.

## Cut Process - Events

This page contains the activities that are performed to cut the service agreement. Open this page using **Credit & Collections**, **Cut Process** and then navigate to the **Events** tab.

### Description of Page

Refer to [How To Perform Common Cut Process Functions](#) for more instructions describing how to use this page.

Refer to the first tab for a description of **Cut Process** and **Cut Process ID**.

The **Cut Events** scroll contains the process's cut events. Refer to [Cut Processes Are Created From Templates](#) for information about how the system defaults a process's events from its template.

- Note, all information in the scroll is protected if the event's Status is **Complete** or **Canceled**.
  - **Event Sequence** is the unique identifier of the event.
  - **Status** defines the [state of the event](#).
  - The **Cancel** button appears if the event is **Pending** or **Waiting**, and the event's **Cut Event Type** has a **Cancel Logic** plug-in. Click this button to cancel the event.
  - The **Cut Event Type** defines the event's activity (e.g., Email sent, a To Do entry is generated, a letter is sent).
  - The remaining fields work in unison:
    - If **Dependent on Other Events** is turned on, the event can only be triggered after the events specified in the **Event Dependencies** grid are all **Complete** or **Canceled**. You use the **Days After** field to define when this event is activated. For example, if you enter 5, this event will be activated 5 days after all of the dependent events are completed / cancelled. If you enter 0 (zero), this event will be activated immediately after the dependent events are completed / cancelled. Refer to [Calendar vs Work Days](#) for a descriptions of how days are counted.
- If all dependent events are already **Completed** or **Canceled**, you cannot specify **Days After**. Rather, you must enter the desired activation date in the **Trigger Date** field.
- If **Dependent on Other Events** is turned off, enter the desired activation date in the **Trigger Date** field. If this field is off, the **Days After** field and the **Event Dependencies** grid are hidden.

**If you can't wait for the Overdue / Cut Event Manager to run.** The [Overdue / Cut Event Manager](#) is a background process that activates events on their trigger date. If you don't want to wait for this process to run, you can click the **Trigger Events** button on the **Main** tab to activate the process's events (that are ready for activation).

## How To Perform Common Cut Process Functions

The topics in this section describe how to perform common cut process maintenance functions. Refer to [The Big Picture Of Cut Processing](#) for high-level information about cut processing.

## Contents

- [How To Create A Cut Process](#)
- [How To Change Cut Events](#)
- [How To Cancel A Cut Process](#)

## How To Create A Cut Process

Cut processes are created when an overdue event is activated. You cannot create an ad hoc cut process.

## How To Change Cut Events

When a cut process is first created, it has one or more cut events. The events are the activities that will be performed to persuade the customer to pay the outstanding debt.

The number and type of events that are created when a cut process is initiated are defined on the cut process's cut process template. The following points describe how to add / change / delete events on a cut process if the defaulted events are not satisfactory.

- Use **Control Central** to choose the account with the cut process whose events need to be changed.
- After the account is populated on Control Central, choose the **Cut Process** option on the account or service agreement context menus to transfer to the cut process transaction in update mode. Note, an account's **Active** cut processes can be selected from the [Credit and Collections zone](#).
- To add a new event, transfer to the [Cut Process - Events](#) tab and press the + button in the **Cut Events** scroll. At this point, the event has not been added to the database; rather, it just exists in memory. Before you add the event to the database, you must specify the following information:
  - Choose an **Event Sequence** so that the new event will be positioned properly in respect of the other events.
  - Choose a **Status** of [Pending](#).
  - Choose the desired **Cut Event Type**.
  - If the activation of the new event is dependent on the successful completion of earlier events, turn on **Dep on Other Event** and then:
    - specify the sequences of the dependent events in **Event Dependencies** and
    - specify the how many days after the completion of the last dependent event that the new event should be triggered.
  - If the activation of the new event is NOT dependent on the successful completion of earlier events, turn off **Dep on Other Event** and use **Trigger Date** to define the date on which the event should be activated (i.e., completed).
- To delete an existing event, click on the event in the tree on [Cut Process - Main](#). This will transfer you to the **Events** tab where you can click the - button to remove the event. At this point, the event has not been removed from the database; rather, it's been removed in memory.
- To change an existing event, click on the event in the tree on [Cut Process - Main](#). This will transfer you to the **Events** tab where you can make the desired changes.
- After all desired changes have been made, save the cut process.

## How To Cancel A Cut Process

The system will cancel a cut process when the [cancel criteria defined on the overdue process are satisfied](#). The following points describe how to manually cancel a cut process.

- Use **Control Central** to choose the account with the cut process to be cancelled.
- After the account is populated on Control Central, choose the **Cut Process** option on the account or service agreement context menu to transfer to the cut process transaction in update mode for the account. A list of all cut processes associated with the account appears. If only one cut process exists, it is automatically selected for you. Note, an account's **Active** cut processes can be selected from the [Credit and Collections zone](#).
- On the **Main** tab, click **Cancel** and answer any prompts.

## Payment Arrangement for Bills

A payment arrangement is an agreement with a customer to payoff severely overdue debt in installments. Bills sent to customers with payment arrangements contain charges for both their current services and their payment arrangement installment amount.

Organizations that practice [open-item accounting](#) who collect on overdue bills may allow customers to pay off their overdue bills in installments by setting up a payment arrangement for one or more bills. This transaction is used to maintain bill-oriented payment arrangements.

Refer to [Bill-Oriented Payment Arrangements](#) for background information.

Open this page using **Credit & Collections, Payment Arrangement for Bill** from the main menu.

### Description of Page

The **SA** (service agreement) **Info** is a concatenation of important details about the payment arrangement service agreement (PA SA). The **SA ID** is the unique identifier of the PA SA. These values only appear after the PA SA exists on the database.

The **Current Balance** that appears beneath **SA Info** contains the PA SA's current balance. When you initially set up a PA SA this balance will be zero. It is only non-zero if the customer has not paid a billed installment.

**Payoff Balance** beneath **SA Info** contains the PA SA's payoff balance. The payoff balance is the total amount to be paid off over the PA SA's life. It is only non-zero after delinquent funds have been transferred to the PA SA. This balance is only displayed when it differs from the **Current Balance**.

Refer to [Current Amount versus Payoff Amount](#) for more information.

**Account ID** is the account associated with the PA SA.

The occurrences of **Current** and **Payoff Balance** beneath **Account** are the [account's](#) respective balances. These values are displayed to help you confirm exactly how much the customer currently owes versus how much they will have to payoff over time (remember, after you transfer funds to a payment arrangement, the amount transferred is reduced from the account's current balance).

The **Unpaid Bills** grid contains a row for each of the account's unpaid bills.

Adjacent to each Unpaid Bill is its **Unpaid Amount** and a check box used to select the bill. Every selected bill's unpaid amount will be transferred to the PA SA when you click the **Create** button.

**Warning!** The system automatically indicates that all unpaid debt should be transferred to the payment arrangement. It does this by checking each bill by default.

**Total Candidates** is the sum of ALL unpaid bills (not just the selected bills).

**New Payoff Balance** contains the PA SA's payoff balance. This is the total amount of debt that will be paid off over the lifetime of the PA SA. This value equals the current payoff balance on the PA SA plus all selected debt from the **Candidate SAs** scroll.

The next fields work in unison.

- If you want to define the number of future bills that the customer has to payoff the selected unpaid amount, enter the number of **Installments**. When you tab out of the field, the system sets the **Arrange Amount** equal to **New Payoff Balance / Installments**.

**Arrange Amount may not add up.** You will note that the **Arrange Amount** is rounded up so that if you multiply the number of installments by the calculated installment amount, the result will be greater than the PA SA's payoff amount. Don't worry - only the actual dollar amount of the PA SA's total debt will be billed by the system. The last installment will be slightly lower than the other installments.

- If you want to define the amount charged on future bills to payoff the unpaid amount, enter the **Arrange Amount**. For example, if the customer agrees to pay off their delinquent debt by paying an additional \$50 on each bill, you'd enter "50" in this field. Note - If you enter a dollar amount in this field and tab out, the system will calculate the number of installments and display the number in the **Installments** field.

Define the **CIS Division** and **SA Type** of the payment arrangement service agreement.

**Payment arrangement SA types.** There is a field on SA type called [Special Role Flag](#). Only those SA types with a role of **Payment Arrangement** may be selected.

Clicking the **Create** (or **Change**) button causes the system to perform the following:

- It creates / updates a payment arrangement service agreement (PA SA). It sets the installment amount on this service agreement equal to the installment amount specified on the page.
- It transfers unpaid debt from the selected bills to the PA SA. The adjustment type used to transfer these funds is defined on the PA SA's SA type.
- It links the "transfer from" adjustments to the unpaid financial transactions on the bills. This will cause the unpaid FT's to be linked to **Balanced** match events. This means that any **Active** Overdue Processes for the bills should **Inactivate** (i.e., cancel) because the bills will no longer be considered unpaid. Refer to [How Are Overdue Processes Cancelled](#) for more information..

- It reduces the current balance on the PA SA by the amount of debt transferred to the PA SA. Why? Because when delinquent debt is transferred to the PA SA, its current balance increases. Because current balance contains the amount the customer currently owes, this balance must be reduced because the customer is going to pay off the debt in installments (and therefore they don't currently owe anything). The adjustment type used to transfer these funds is defined on the PA SA's SA type.

Clicking **Break** causes the PA SA to become broken. Please refer to the note below for what transpires when a PA is broken.

**Break logic is in a plug-in.** Please be aware that the logic that is executed when a payment arrangement is broken exists in a plug-in algorithm (plugged-in on the account's [Collection Class Overdue Rules](#)). The base package algorithm performs many functions (refer to [Breaking A Bill Oriented Payment Arrangement](#) for the details). The bottom line is that when a PA SA is broken, the originating bills will have their debt reinstated and this will again be subject to the [Overdue Monitor](#). Because this logic is in a plug-in, you can develop alternative logic and plug it in if the base package logic is not satisfactory.

Clicking **Cancel** causes the PA SA to become canceled. 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.

## Contents

- [How To Set Up A Payment Arrangement](#)
- [How To Add Additional Debt To A Payment Arrangement](#)
- [How To Break A Payment Arrangement](#)
- [How To Cancel A Payment Arrangement](#)

## How To Set Up A Payment Arrangement

To set up a new payment arrangement:

- Open [Payment Arrangement for Bills](#) in add mode.
- Select the **Account** in question.
- Select the debt to be transferred to the payment arrangement (by turning on/off the checkboxes adjacent to the **Unpaid Bills**).
- Enter the number of **Installments** or the **Arrange Amount**.
- Define the **CIS Division** and **SA Type** of the new payment arrangement service agreement.
- Click the **Create** button.

## How To Add Additional Debt To A Payment Arrangement

To transfer additional debt to an existing payment arrangement:

- Open [Payment Arrangement for Bills](#) in update mode.
- Select the existing payment arrangement in the search.

- Select the debt to be transferred to the payment arrangement (by turning on/off the checkboxes adjacent to the **Unpaid Bills** amounts).
- Enter the number of **Installments** or the **Arrange Amount**.
- Click the **Change** button.

**After transfer.** After transferring the funds to the payment arrangement service agreement, confirm the account's payoff and current balances make sense (the payoff balance is the total amount of debt that the account will eventually have to payoff, the current balance is how much they currently owe. The account's payoff and current balances are displayed in the page's second section. If you do not understand the difference between payoff balance and current balance, refer to [Current Amount versus Payoff Amount](#).

## How To Break A Payment Arrangement

You would break a payment arrangement when a customer doesn't make the agreed payments. Typically, your overdue processes will break a payment arrangement behind-the-scenes if a customer doesn't pay. Refer to [Breaking A Bill Oriented Payment Arrangement](#) for information about how the system monitors payment arrangements.

If you want to manually break a payment arrangement,

- Open [Payment Arrangement for Bills](#) in update mode.
- Select the existing payment arrangement in the search.
- Click the **Break** button.

## How To Cancel A Payment Arrangement

You would cancel a payment arrangement if you made a mistake and never intended to set up the payment arrangement in the first place. To cancel a payment arrangement

- Open [Payment Arrangement for Bills](#) in update mode.
- Select the existing payment arrangement in the search.
- Click the **Cancel** button.

# Rebate Claims

The Rebate Claims feature allows you to search, view, and maintain rebate claims submitted as part of a conservation program.

For more information about conservation programs, see [Conservation Programs](#).

## Contents

- [Rebate Claim Query Portal](#)
- [Rebate Claim Portal](#)

## Rebate Claim Query Portal

You use the Rebate Claim Query Portal to search for rebate claims. Use the following procedure to search for rebate claims:

- Select **Main Menu, Customer Information, Rebate Claim** to open the Rebate Claim Query portal.
- Enter search criteria in the Rebate Claim Search zone and click **Refresh**. You can search by:
  - Customer Name
  - Account ID
  - Conservation Program
  - Start and End Dates
- A list of rebate claims matching the search criteria appears, displaying the following for each:
  - Claim Date
  - Rebate Claim (Status, Claim Date, Conservation Program, and Total Amount)
  - Account (Name, Customer Class, Current Balance, and Account ID)
  - Total Rebate Amount
- To view a rebate claim, click the **Rebate Claim** link. The selected rebate claim opens in the Rebate Claim portal.

## Rebate Claim Portal

You use the Rebate Claim portal to view, edit, and maintain rebate claims. To add a rebate claim, open **Main Menu, Customer Information, Rebate Claim**.

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

## Contents

- [Rebate Claim Zone](#)
- [Adding Rebate Claims](#)

[Rebate Claim Line Zone](#)  
[Rebate Claim Log Zone](#)

## Rebate Claim Zone

You use the Rebate Claim zone to view the details of rebate claims. The Rebate Claim zone opens when you first create a new rebate claim, or when you select a rebate claim on the Rebate Claim Search zone. The Rebate Claim zone displays the following information about the selected rebate claim:

- Basic Information: information about the claim, including:
  - Conservation Program
  - Status (Pending, Issue Detected, Approval in Progress, Receipt Verification, Complete, or Canceled)
  - Rebate Claim number
  - Claim Date
  - Total Rebate Amount
- Customer Details: details about the customer who submitted the claim, including:
  - Person
  - Account
  - Premise
  - Mailing Address
- Contact Details: contact information for customer who submitted the claim, including:
  - Name
  - Phone Number
  - Fax Number
  - Email Address
- Premise Information: information about the physical premise of the customer who submitted the claim, including:
  - Premise Property Type
  - Residential Property Type
  - Building Vintage
- Payment Release Authorization Details: details about the person to whom the rebate payment should be made, including:
  - Authorized By
  - Authorization Date
  - Payee Name
  - Phone Number
  - Mailing Address

- Issue Details: issues related to the rebate claim, including manually submitted issues as well as those detected through the Verify action.
- Approval Details: details related to users assigned to approve the rebate claim, including:
  - Current Approver
- Reject Details: details related to why a claim was rejected, including:
  - Reject Reason
  - Date Rejected
  - Rejected By

To view a different rebate claim, select the rebate claim from the **Select a Rebate Claim** drop-down list and click **Go**.

## Contents

- [Rebate Claim Status](#)
- [Rebate Claim Actions](#)
- [Editing Rebate Claims](#)
- [Verifying Rebate Claims](#)
- [Approving Rebate Claims](#)
- [Rejecting Rebate Claims](#)
- [Noting Issues Related to a Rebate Claim](#)
- [Canceling Rebate Claims](#)

## Rebate Claim Status

Rebate claims are assigned a status to indicate the current state of the claim within the system. Valid statuses include:

- **Pending** indicates the claim is pending. This is the initial state of a rebate claim when first created.
- **Issue Detected** indicates that one or more issues have been detected with the claim. Issues can include those manually created as well as issues detected through the Verify action.
- **Approval in Progress** indicates the claim is awaiting approval.
- **Receipt Verification** indicates that receipts associated with the rebate claim's claim lines need verification.
- **Complete** indicates the claim was approved and is complete.
- **Rejected** indicates the claim was rejected.
- **Canceled** indicates the claim has been canceled.

## Rebate Claim Actions

You can perform a number of actions on a rebate claim, including:

- **Edit:** Used to edit the details of the rebate claim.
- **Verify:** Used to verify the supplied details of the rebate claim, and change its status to either Approval in Progress (if the claim passes the verification) or Issue Detected (if any issues are detected). This helps ensure the rebate claim has been properly set up.

- **Approve:** Used to approve a submitted rebate claim, and change its status to Complete or Receipt Verification (if receipts had not been previously verified).
- **Reject:** Used to reject a submitted rebate claim, and change its status to Complete.
- **Issue Detected:** Used to note an issue related to the rebate claim, and change its status to Issue Detected.
- **Cancel:** Cancels a rebate claim, changing its status to Canceled.

The actions available are based on the current status of the rebate claim. The table below summarizes the actions available at each status.

Status	Valid Action
Pending	Edit, Verify, Issue Detected, Cancel
Issue Detected	Edit, Verify, Cancel
Approval in Progress	Approve, Reject, Issue Detected
Receipt Verification	Issue Detected
Completed	None
Rejected	Issue Detected
Canceled	None

## Editing Rebate Claims

Use the following procedure to edit a rebate claim:

- Click **Edit**.
- Edit the details of the rebate claim as needed.
- Click **Save**.

## Verifying Rebate Claims

To verify a rebate claim, click **Verify**.

## Approving Rebate Claims

To approve a rebate claim, click **Approve**.

## Rejecting Rebate Claims

Use the following procedure to reject a rebate claim:

- Click **Reject**.
- Select the **Reject Reason** on the Reject Rebate Claim dialog.
- Click **OK**. To close the dialog without rejecting the claim, click **Cancel**.

## Noting Issues Related to a Rebate Claim

Use the following procedure to note an issue related to a rebate claim:

- Click **Issue Detected**.
- Enter details about the issue in the Comments field of the Issue Comments dialog.
- Click **OK**. To close the dialog without saving the comment, click **Cancel**.

## Canceling Rebate Claims

Use the following procedure to cancel a rebate claim:

- Click **Cancel**.
- Select the **Cancel Reason** on the Cancel Rebate Claim dialog.
- Click **OK**. To close the dialog without cancelling the rebate claim, click **Cancel**.

## Adding Rebate Claims

You can also add rebate claims from the Rebate Claim Query portal. Use the following procedure to add a rebate claim:

- Select **Main Menu, Customer Information, Rebate Claim +** or click **Add** on the title bar of the Rebate Claim Search zone. The user is navigated to Control Central to select the customer filling the rebate claim. Click **Continue** once the correct account is in context.
- Select the Conservation Program associated with the rebate claim and click **OK**.
- Complete the Rebate Claim details, including:
  - Claim Date for the rebate claim
  - Customer Details
  - Contact Details
  - Premise Information
  - Payment Release Authorization Details
- Click **Save**. The rebate claim is displayed on the Rebate Claim portal.
- Edit the claim as needed in the Rebate Claim zone.
- Add claim lines to the claim using the Rebate Claim Line zone.

## Rebate Claim Line Zone

You use the Rebate Claim Line zone to add, view, and edit rebate claim lines associated with a rebate claim. A rebate claim can have one or more claim lines, one for each item for which a rebate is requested. For example, if a customer purchases new energy efficient appliances (such as a washing machine and dryer), the rebate claim they submit would have a rebate claim line for each appliance. The Rebate Claim Line zone displays the following details for each claim line:

- Rebate Definition
- Manufacturer/Model
- Rebate Amount

- Icons to Edit, Delete, and Verify claim lines

## Contents

- [Adding Rebate Claim Lines](#)
- [Editing Rebate Claim Lines](#)
- [Deleting Rebate Claim Lines](#)
- [Verifying Rebate Claim Lines](#)

## Adding Rebate Claim Lines

Use the following procedure to add a rebate claim line to a rebate claim:

- Click **Add Rebate Claim Line** in the title bar of the Rebate Claim Line zone.
- Select the **Service Agreement** associated with the claim line from the drop-down list.
- Select the **Rebate Definition** for the claim line from the drop-down list. This list is filtered based on the Conservation Program associated with the rebate claim.
- Select the **Manufacturer/Model** for the claim line from the drop-down list. This list is filtered by the Rebate Definition selected.
- Enter the **Install Date** for the claim line.
- Enter the **Number of Units** being claimed.
- Click **Save**. To return to the Rebate Claim Line zone without adding the claim line, click **Cancel**.

## Editing Rebate Claim Lines

Use the following procedure to edit an existing rebate claim line on a rebate claim:

- Click the edit icon for the claim line you wish to delete.
- Edit the details of the claim line as appropriate.
- If you are revising the data based on updated information, check the **Data Revised** checkbox and select the reason for the revision from the **Data Revised Reason** drop-down list. The **Original Data Details** section displays the data originally entered for the claim line.
- If the claim line contained bad data, check the **Bad Line** checkbox and select the reason for the revision from the **Bad Line Reason** drop-down list. The **Original Data Details** section displays the data originally entered for the claim line.
- Click **Save**. To return to the Rebate Claim Line zone without changing the claim line, click **Cancel**.

**Note:** You can edit a rebate claim line only when the rebate claim is in Pending status. In addition, you cannot edit a rebate claim line after it has been verified.

## Deleting Rebate Claim Lines

Use the following procedure to delete a rebate claim line from a rebate claim:

- Click the delete icon for the claim line you wish to delete.

- Click **OK** on the Confirm Delete dialog. To close the dialog without deleting the claim line, click **Cancel**.

**Note:** You can delete a rebate claim line only when the rebate claim is in Pending status.

### Verifying Rebate Claim Lines

Use the following procedure to verify a rebate claim line on a rebate claim:

- Click the **Verify** button for the claim line you wish to verify.
- Verify that the details of the claim line match the submitted receipts.
- Click **Verify**. To return to the Rebate Claim Line zone without adding the claim line, click **Cancel**.

**Note:** You cannot edit a rebate claim line after it has been verified.

### Rebate Claim Log Zone

This is a standard [log zone](#). You use the Rebate Claim Log zone to view logging details associated with a rebate claim.

### Adding Rebate Claim Log Entries

Use the following procedure to add a rebate claim log entry:

- Click **Add Log Entry** in the title bar of the Rebate Claim Log zone.
- Enter the log entry details in the Add Rebate Claim User Log dialog.
- Click **OK**. To close the dialog without saving the log entry, click **Cancel**.