

Oracle Utilities Customer Care and Billing

Business Process Guide

Volume 2

Release 2.3.1

E18367-01

September 2010

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.

Payments

In this section, we describe how to manage your customer's payments.

Contents

- [The Big Picture of Payments](#)
- [Maintaining Payment Events](#)
- [Payment Event Quick Add](#)
- [Payment Quick Add](#)
- [Maintaining Payments](#)
- [Payment Portal](#)
- [How To](#)
- [Financial Transactions On A Payment](#)
- [Payment History](#)
- [Payment / Tender Search](#)
- [Payment Event Exception](#)
- [Payment Exception](#)
- [Maintaining Deposit Controls](#)
- [Maintaining Tender Controls](#)
- [Interfacing Payments From External Sources](#)

The Big Picture of Payments

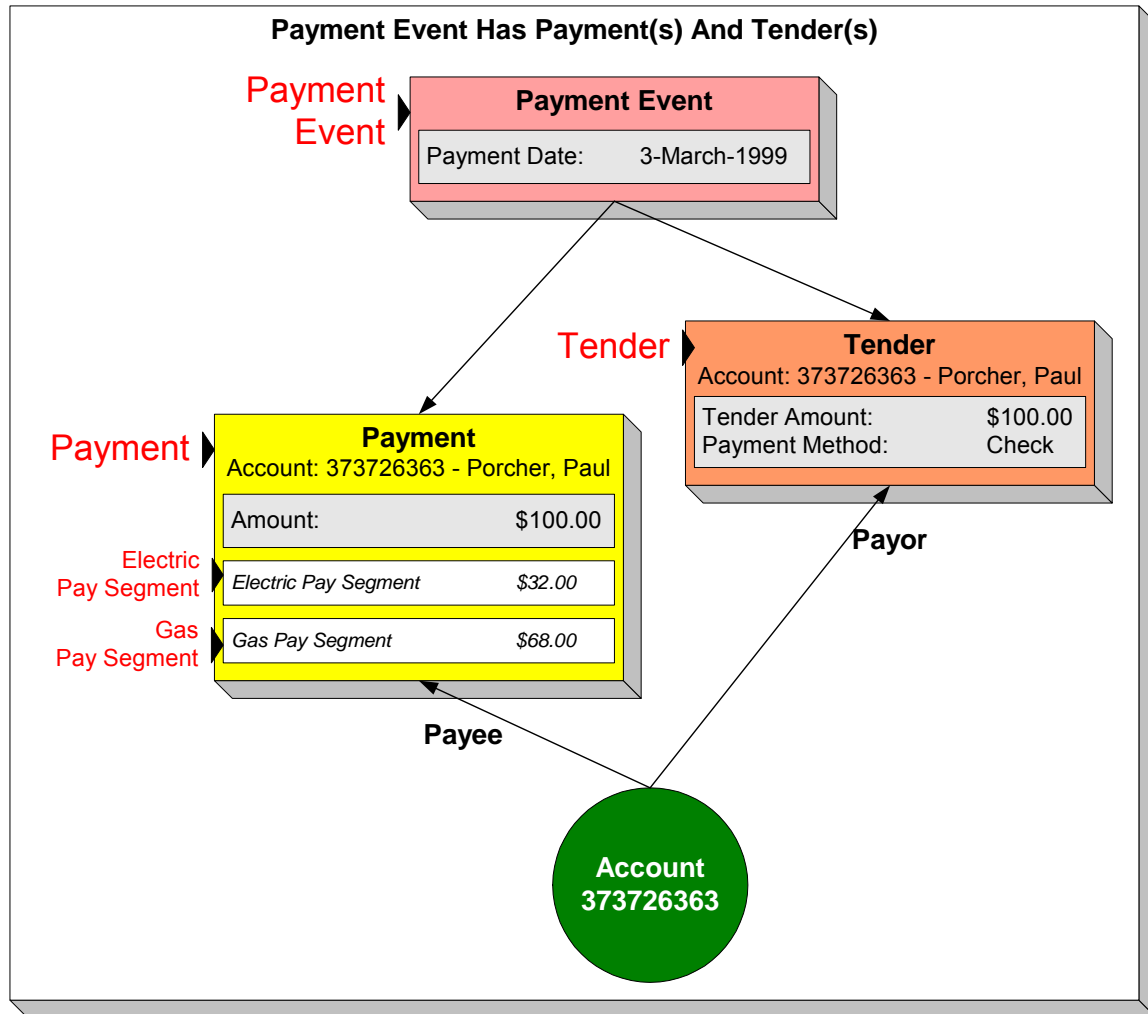
A payment reduces how much an account owes. The topics in this section provide background information about a variety of payment topics.

Contents

- [A Payment Event Has Payments And Tenders](#)
- [Multiple Tenders Used To Pay For Multiple Accounts](#)
- [An Overview Of The Payment Event Creation & Allocation Process](#)
- [Distributing A Payment Event](#)
- [Distributing A Payment Amongst An Account's Service Agreements](#)
- [Overpayment](#)
- [Canceling A Tender Versus Canceling A Payment](#)
- [NSF Cancellations](#)
- [Transferring A Payment](#)
- [Unbalanced Payment Events](#)
- [How And When Payments Appear On Bills](#)
- [Tender Management and Workstation Cashiering](#)
- [Exceptions](#)
- [Payment Financial Transaction Considerations](#)
- [A Payment May Affect More Than Just Customer Balances](#)
- [Automatic Payments](#)
- [Issuing A Payment Advice Instead Of Creating An Automatic Payment](#)

A Payment Event Has Payments And Tenders

The explanation in [The Financial Big Picture](#) provides an accurate, but incomplete view of payments. The missing pieces concern *payment events* and *tenders*. The following diagram illustrates the difference between a payment event, its payment(s) and its tender(s).



The following concepts are illustrated above:

A payment event defines the event

A payment event is required whenever any form of payment is received. The payment event defines the payment date (and that's all).

A payment event has tender(s)

A tender exists for every form of tender remitted as part of the payment event. A payment event must have at least one tender otherwise nothing was remitted. A payment event may have many tenders when multiple payment methods are associated with an event (e.g., paying with cash, a check, and a credit card).

A payment is allocated to account(s)

The total amount of tenders under a payment event is distributed to one or more accounts.

A payment is distributed to SA's

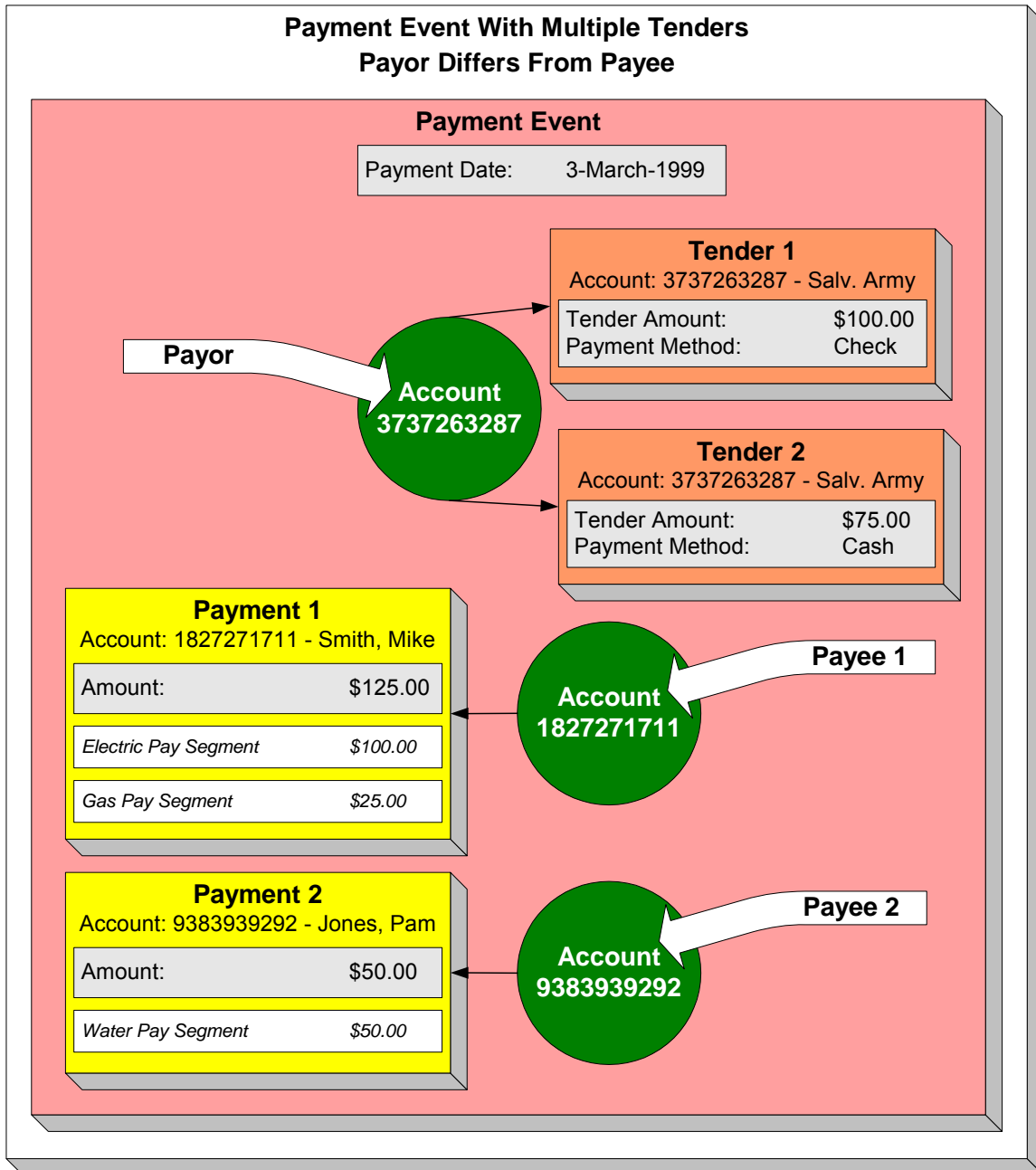
The system allocates an account's payment amount amongst its service agreements. The system creates a payment segment for each service agreement that receives a portion of the payment.

Payor and payee are frequently the same

The account remitting the tender (the payor) is frequently the same as the account to which the funds are allocated (the payee). The next illustration provides an example when this is not the case.

Multiple Tenders Used To Pay For Multiple Accounts

The following diagram illustrates a payment event with multiple tenders where the payor of the tender is not the same as the account(s) receiving the payment.

**A payment event may have many tenders**

A single payment event may have many tenders. While the above example shows both tenders being paid by the same account, each tender may reference a different account.

Many accounts may be paid under 1 event

The total amount of tenders under a payment event are distributed to one or more accounts.

Payor may differ from payee

The account(s) remitting the tender may differ from the account(s) whose debt is relieved.

An Overview Of The Payment Event Creation & Allocation Process

When a payment event occurs, the system stores a tender for each form of remittance (e.g., cash, check, charge). It then allocates the sum of the tenders to one or more accounts.

By default, the system allocates the sum of the tenders to the account that remits the tenders. You may override this default and specify any number of accounts and their respective payment allocation amount. This is useful, for example, when a social service agency pays for many accounts. If applicable, you may also configure the system to use your own payment event distribution rule(s).

Refer to [Distributing A Payment Event](#) for more information.

The system distributes a payment amongst an account's service agreements based on the age of each service agreement's debt AND distribution priority. The system creates a payment segment for each service agreement that receives part of the payment.

Refer to [Distributing A Payment Amongst An Account's Service Agreements](#) for more information.

You may manually redistribute the payment amount amongst the account's service agreements before you commit the distribution. When the distribution is acceptable, you freeze the payment. Freezing a payment causes the system to create a financial transaction for each related payment segment. It is the financial transaction(s) that causes the service agreements' payoff and current balances to be reduced. The financial transaction also contains the journal details that debit "cash" and credit some other GL account.

And that's it. The remaining topics in this section provide more information about the creation and allocation of payment events.

Batch and real-time payment event creation / allocation. There is only one payment event creation / allocation routine and therefore anything the batch payment process does for whole batches of payments, you can do to a payment on-line.

Distributing A Payment Event

Warning! This section deals with the concept distributing a payment amount into payment(s). It does not discuss the distribution of a single payment into segments. **For more information** about payment distribution, refer to [Distributing A Payment Amongst An Account's Service Agreements](#).

The base-package, by default, creates a single payment for a payment event. Some business practices require potentially many payments to be created when payment events are added.

A few examples of when multiple payments may be necessary are:

- A payment amount needs to be distributed towards different distribution types:
 - \$50 in interest
 - \$60 in collection charges

- \$70 in taxes

Each of the above distributions is realized as a separate payment identified by its own match type (i.e., there'll be one match type called Interest, another called Collection Charges, etc.).

In this example, the debt of a single service agreement may be relieved by each of these payments.

- In a similar way, you may want to create a separate payment for an overpayment to differentiate it from regular payments (using yet another match type).
- In the case of a social service agency that pays for many accounts, a single payment event may be distributed amongst multiple accounts.

The method by which a payment amount is distributed to create payment(s) is contained in [Create Payment algorithms](#) plugged in on a distribution rule.

There is yet another aspect to having control over how payment events are created. The default method of creating payment events assumes knowledge of account IDs (of the payor and the payee) when making a payment. In cases where payments are made by and towards business entities other than accounts, knowledge of their corresponding account IDs may not be available at payment time. Consider the following examples:

- A payment is made to pay a person's bill, providing the person's identification at payment time.
- A payment is made towards a specific service agreement using a "contract number" defined as SA characteristic.

The method by which the tender account is determined by means of an alternate identifier is contained in [Determine Tender Account algorithm](#) plugged in on a distribution rule.

Refer to [Making Payments Using Distribution Rules](#) for information on how to configure your system to use this distribution method.

Distributing A Payment Amongst An Account's Service Agreements

Warning! This section deals with the concept of distributing a payment amongst an account's service agreements. It does not discuss how the sum of a payment event's tenders is balanced out by payment allocations. **For more information** about payment event balance, refer to [Unbalanced Payment Events](#).

A payment must be distributed to one or more service agreements for its financial impact to be realized. When a payment satisfies an account's entire debt, you don't have to worry about how the system distributes the payment. The concept of payment distribution is only relevant when a partial or excess payment is distributed.

The first important point to understand is that the method of distributing a payment amongst an account's service agreements is contained in an algorithm that's plugged in on to [Customer Class](#). This means that you can have different distribution algorithms for different customer classes.

Manual overrides. Most of the time, you'll let the payment distribution algorithm distribute the payment amongst an account's service agreements. However, you may manually distribute a payment when a customer directs a payment to specific service agreement(s).

The following explanation describes one of the base package payment distribution algorithms (refer to [Payment Distribution – Pay Priority and Debt Age](#) for information about this algorithm). This algorithm distributes a payment based on:

- The age of each service agreement's debt.
- The payment distribution priority of each service agreement's SA type.

The following diagram helps illustrate how the distribution algorithm works.

Important! There are other payment distribution algorithms in the base package. Click [here](#) to see the available algorithm types.

Distributing A Payment Amongst An Account's Service Agreements Is Controlled By The Age Of The Debt And The SA's Distribution Priority					
		Priority 10 (Highest)		Priority 90 (Lowest)	
		SA #1	SA #2	SA #3	
Excess Credits		16			
New Debits*		13	14	15	
Non-Delinquent		10	11	12	
A r r e a r s	35-day Arrears	5	6	9	35-day Arrears
	66-day Arrears	3	4	8	52-day Arrears
	93-day Arrears	1	2	7	93-day Arrears
*--Includes all charges not yet invoiced to the customer.					

The above example shows three columns, one for each service agreement linked to a hypothetical account. Notice that two of the service agreements have the same distribution priority, the third has a lower priority. The numbers in the cells indicate the order in which the system distributes a partial payment.

Debt terminology. Before we can discuss the distribution algorithm, you must understand the terminology we use to categorize debt. **Delinquent** debt is associated with financial transactions that appear on overdue bills. **Non-delinquent** debt is associated with financial transactions that appear on current bills. **New debits** debt is associated with financial transactions that do not yet appear on a completed bill (e.g., a late payment charge that hasn't been billed yet).

The following points describe the algorithm used to distribute the partial payment:

- The system pays off delinquent debt of the highest priority service agreements first. In the above example, where multiple service agreements have the same distribution priority, the system does NOT payoff one service agreement before it starts on the next (which one would it pick?). Rather, it distributes the payment amongst the service agreements based on the age of the respective debt on each service agreement. In the above example, this is represented by steps 1 through 6 (notice how the distribution jumps between SA1 and SA2).
- After all delinquent debt has been relieved from the highest priority service agreement(s), the system pays off the next priority until all delinquent debt is relieved. In the above example, this is represented by steps 7 through 9.
- The system next pays off non-delinquent debt using each service agreement's respective distribution priority. Note well, the payment distribution algorithm doesn't associate an age with non-delinquent debt and therefore the distribution is based purely on the service agreements' respective distribution priority. In the above example, this is represented by steps 10 through 12.
- After all non-delinquent debt is relieved, the system next pays off "new debit" debt based on the service agreements' respective distribution priority. In the above example, this is represented by steps 13 through 15.
- Refer to [Overpayment](#) for a description of what happens if money still exists after the above distribution is complete.

Payment segments and financial transactions. A payment segment exists for each service agreement that receives a portion of a payment. Linked to each payment segment is a financial transaction. It is the financial transaction that causes the service agreement's debt to be relieved and the general ledger to be impacted.

Refer to [Payment Exception](#) for more information about how the system handles errors detected during the payment distribution process.

Overriding the distribution algorithm for uploaded payments. The standard distribution algorithm is used for payments that are [Interfaced From An External System](#) unless you specify a MATCH_VALUE and MATCH_FLG on the [Payment Staging](#) row associated with the uploaded payment. These fields are used in conjunction to indicate that the distribution of the payment should be restricted in some way (i.e., the standard payment distribution algorithm should not be used). MATCH_FLG indicates how the payment should be distributed (e.g., only distribute to a specific service agreement), MATCH_VALUE contains the ID of the restriction (e.g., the SA ID).

Open item customers. For an [open-item customer](#), you MUST override the standard distribution algorithm because the payment is distributed as per the open items that it is relieving. Refer to [Payments And Match Events](#) for more information.

Overpayment

Overpayment refers to the situation where money is left over after a payment has been distributed to all eligible service agreements, and all debt is relieved. Refer to [Overpayment Segmentation](#) for a description on how to configure the system to handle your overpayment requirements.

Canceling A Tender Versus Canceling A Payment

A payment event has tender(s) and payment(s). You can cancel a tender when it's not valid, e.g., when a check bounces. You can cancel a payment when the account should not have received the payment (e.g., a misdistribution or a canceled tender).

When you cancel a tender, the system automatically cancels ALL **frozen** payments. We do this because if the tender is canceled, there are no funds to distribute to accounts (unless there are other non-canceled tenders under the event). However, when you cancel a payment, the system does NOT cancel the tender(s) because we assume that, if the tenders were incorrect, you would have canceled them rather than the payment.

NSF Cancellations

When a tender is canceled, a cancellation reason must be supplied. If the cancellation reason indicates a NSF (non sufficient funds) charge should be levied, the system invokes the NSF charge algorithm specified on the tender's account's [customer class](#). Algorithms of this type will typically create an adjustment or billable charge to levy the NSF charge. Refer to [NSFC-DFLT](#) for an example of such an algorithm.

The payor gets the NSF adjustments. It's possible for the payor of a tender to differ from the payee (e.g., when a social service agency pays for other customers' debt). When you cancel a tender you must specify a cancellation reason. If the cancellation reason indicates an NSF charge should be levied, the sample algorithm provided with the system levies the NSF charge on the PAYOR's account. Because adjustments must be linked to a service agreement, the algorithm picks the payor's service agreement with the highest payment distribution priority and levies the adjustment specified in the service agreement's SA type's NSF adjustment type.

Besides calling this algorithm, a NSF cancellation may affect the tendering [account's credit rating and cash-only score](#). The cancellation reason indicates the extent to which the account's ratings are affected.

And finally, whenever a tender is canceled due to non-sufficient funds, the account is scheduled for review by the [Account Debt Monitor](#).

Transferring A Payment

If the account on an event's tender and payment are wrong, you can use the [Transfer](#) button on the Payment Event page to transfer the payment to another account.

If the account on the payment is wrong – but the tender is correct, you can use the [Transfer](#) button on the Payment page to transfer the payment only to another account.

Unbalanced Payment Events

The system, by default, distributes the sum of a payment event's tenders to the account that remits the tender with a single payment. After distribution the sum of the tenders equals the sum of the payments when the event is first created. We refer to such an event as being **balanced**.

However, it is possible for an event's tender amount to not equal the sum of the payment allocations (i.e., the event becomes **unbalanced**). How? Well, there are several ways this can happen:

- While the system DEFAULTS the payment amount to be the tender amount, you can override the payment amount and therefore make a previously balanced event **unbalanced**.
- While the system DEFAULTS the payment account to be the tender account, you can add additional accounts / amounts and therefore make a previously balanced event **unbalanced**.
- When you cancel a tender (e.g., because a check bounces), the system cancels ALL payments linked to the tenders payment event. If the payment event has multiple tenders, this will cause the event to become **unbalanced**. To correct this situation, you must add payment allocations to equal the amount of uncanceled tenders.
- If you cancel a payment and forget to add another payment for the same amount, the event becomes **unbalanced**. To correct this situation, you must add another payment (or cancel the tender).
- You may delete a tender from an event while its tender control is **open**. If you delete a tender and don't do anything about the related payments, the event becomes **unbalanced**.
- You may add a tender to an event at any time. If you don't allocate the tender amount to an account, the event becomes **unbalanced**.

Refer to [Payment Event Exceptions](#) for more information about how the system reminds you about unbalanced payment events.

How And When Payments Appear On Bills

Refer to [Financial Transactions Created Between Bills](#) for more information.

Tender Management and Workstation Cashiering

When you add a tender, you must identify its **Tender Source**. For example,

- A specific cash drawer ID is the source of tenders remitted to a cashier.
- The notional lockbox ID is the source of tenders interfaced from a lockbox.
- The remittance processor is the source of tenders interfaced from a remittance processor.

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

For more information, refer to [Setting Up Tender Sources](#).

A tender source's tenders must be balanced against an expected total before they can be deposited at a bank. This periodic balancing requires all tenders to exist in respect of a **Tender Control**. Over time, a tender source may have many tender controls (one per balancing event).

An example of a cashier's cash drawer will help clarify the tender control concept:

- When a cashier starts in the morning, s/he starts with a fresh cash drawer (i.e., one without tenders). Whenever a drawer starts afresh, a new Tender Control must be created because you balance the contents of a drawer.
- A cash drawer typically contains funds to make change. These funds are the tender control's **Opening Balance**.

Default note. A tender control's starting balance defaults from its tender source.

- During the day, customers remit tenders to the cashier. Every tender put into the drawer is associated with the drawer's tender control created at the start of the day.
- A tender control's balance increases during the day as tenders are recorded. A cashier can view the balance at any time.
- The cashier can turn in funds to the head cashier during the day. Each turn in event can be recorded in the system. Note well, if the amount of funds in a tender control exceeds the maximum balance defined on the tender control's tender source, a warning is issued to the cashier to remind him/her to turn in funds.
- At some point, the contents of the drawer must be balanced against the total tenders linked to the tender control. When balancing starts, no additional tenders may be put into the tender control. If the cashier receives additional tenders after balancing starts, a new tender control must be created (and the above process starts afresh).
- During the balancing process, some modifications may be made to the tenders associated with the tender control, but no additional tenders may be added. When the tender control is balanced, neither it nor its tenders may be modified.

For more information, refer to [Managing Your Cash Drawers](#).

While the above explanation is true, it isn't complete. In addition to the requirement that a tender must reference a tender control, the tender control must refer to a **Deposit Control**. Deposit controls give you administrative control over all of the tender controls whose contents will be deposited en masse. The following concepts will help explain the power of deposit controls:

- As explained above, when a cash drawer is started afresh, a new tender control must be created. The tender control "holds" all new tenders received by the cashier.
- Similarly, when a tender control is created, it must reference a deposit control. During the day, a deposit control's tender controls are constantly changing. You can view the total impact of a deposit control's tender controls at any time.
- At some point, you will want to deposit the tenders received during the day. To do this, you must indicate how much will be deposited at the bank. This deposit amount must equal the sum of the tender controls linked to the cash drawer. When this deposit balancing starts, no additional tender controls may be associated with the deposit control.

- When the deposit total equals the sum of the tender controls, the deposit control becomes **balanced** and no changes may be made to it, its tender controls, or its tender controls' tenders.

For more information, refer to [Managing Your Cash Drawers](#).

Background processes use the same concepts. Tenders that are interfaced from external sources (e.g., lockboxes and remittance processors) make use of the concepts describe above. For example, tenders interfaced from a remittance processor are linked to a tender control and this tender control is linked to a deposit control. The main difference is that the background processes require no human intervention; the system automatically creates tender and deposit controls and sets their states to **balanced** when the interface concludes successfully.

The ACH activation process also creates tender and deposit controls. Refer to [Activating Automatic Payments](#) for more information.

The topics in this section elaborate on the tender management concepts described above.

Contents

- [Managing Your Cash Drawers](#)
- [Turn Ins](#)
- [Balancing By Tender Type](#)
- [Cash Back](#)
- [Managing Payments Interfaced From External Sources](#)

Managing Your Cash Drawers

Warning! This section assumes you are familiar with the concepts described in [The Lifecycle Of A Deposit Control](#) and [The Lifecycle Of A Tender Control](#).

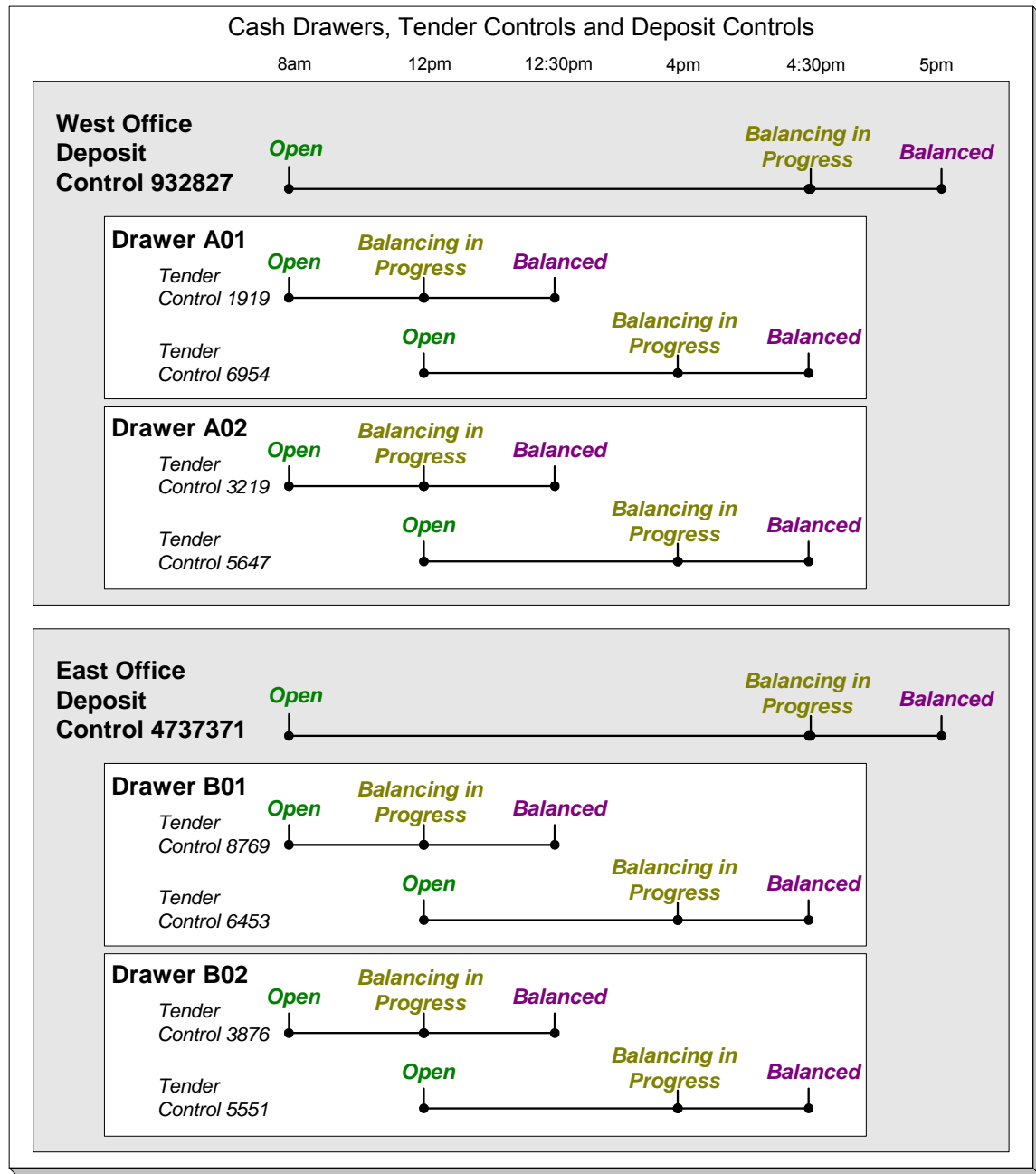
There are many ways to handle the daily management of tenders received via cash drawers. It really depends on how your organization works. To help you understand the potential of the system, we'll continue the example started above.

Assume that the cash drawers in your western office are balanced and deposited independently from those in your eastern office. We'll assume that both offices follows the same daily routine:

- Load fresh drawers first thing in the morning. Each drawer contains a starting balance of \$150.00. Note: the drawer's tender control's starting balance defaults from its tender source.
- At 10 am, the cashier turns in funds to the chief cashier and continues to receive additional tenders.
- At 12 noon, each drawer is pulled and balanced by a supervisor.
- By 12:30 pm, the tender controls are balanced.
- At 4 pm, the cashiering stations are closed. Each drawer is pulled and balanced by a supervisor.
- By 4:30 pm, the tender controls are balanced.

- At 5 pm, the deposit control is balanced and funds are ready to be deposited at the bank.

Given this, the following diagram illustrates the deposit controls and tender controls used by each office on a given day.



The following concepts are illustrated above:

- An **Open** deposit control must exist before you can create a tender control. And an **Open** tender control must exist before you can create a tender. From a business process standpoint, this means:
 - A supervisor would create a deposit control at the start of the day (8 am in the above illustration).

- Each cashier would create a tender control when they start a drawer and reference the deposit control created by the supervisor.
- During the day, the cashier can turn-in moneys to the chief cashier. These turn-in events are recorded in the system as they play a part in the ultimate balancing of the drawer. Refer to [Turn Ins](#) for more information.
- At some point, the contents of a drawer can be pulled and balanced. If additional tenders can be received in a drawer, a new tender control must be created for the drawer. Refer to [Balancing By Tender Type](#) for more information.
- At the end of the day, the supervisor checks to make certain that all tender controls linked to the deposit control are **Balanced**. After this has been done, the supervisor indicates the deposit amount on the deposit control and changes it to **Balanced**. Notice that in the above illustration each deposit control references four tender controls.

Alternate currency payments. If your organization accepts payments in alternate currencies, that is, a currency other than the account's currency; a supervisor would create a deposit control for each such currency. Likewise, the cashier would create a tender control for each such currency.

Refer to [Alternate Currency Payments](#) for more information.

- Typically, a cash drawer has one tender control **Open** at any point in time (meaning that the tenders being received are being linked to a specific tender control). However, this is not a hard rule. If you want, you may have multiple tender controls **Open** at any point for a specific cash drawer (for example, if multiple cashiers can work the same drawer during the day but take their drawer with them).
- Typically, a specific cashier puts tenders into a specific tender control. However, this is not a hard rule. On a tender control, you can define if it's limited to a specific operator OR if any operator can link tenders to it.
- When you're ready to balance a drawer, you change the tender control to **Balancing in Progress**. This prevents new tenders from being added to the tender control. If the cashier can continue to receive tenders, s/he must create another tender control. In the above example, all drawers are balanced at 12 noon by a supervisor while the cashier continues to take payments.
- When the tender control is balanced, you change its state to **Balanced**. This prevents any changes to the tender control or its tenders.
- All tender controls exist in respect of a deposit control (in fact, the deposit control must be created before the tender control). This way, a supervisor can check the state of the related drawers throughout the day. Notice that the state transition of a deposit control is identical to that of the tender control (refer to [The Lifecycle Of A Deposit Control](#) and [The Lifecycle Of A Tender Control](#)). There is only a temporal difference. Notice that the deposit control stays open throughout the day while any number of tender controls are being opened and balanced.

Multiple deposits in a day. While the above example illustrates a single deposit per office per day, it is quite possible to have multiple deposit controls on any given day.

Turns ins. The above example did not illustrate the fact that a cashier can turn-in moneys during the day without having to balance the drawer. Refer to [Turn Ins](#) for more information.

Turn Ins

A cashier may optionally turn-in funds received into a cash drawer to a head cashier. The turn-in process requires two steps:

- Each time a cashier turns-in funds, they add a turn-in event on the [Tender Control – Turn Ins](#) page. Note that a separate turn-in event is required for each type of tender that's turned in. This is because the balancing of a tender control is performed for each tender type and therefore the system must know how much of each tender type has been removed from the drawer.

Turn in warning. If the amount of cash-like funds in a tender control exceeds the maximum balance defined on the tender control's tender source, a warning is issued to the cashier to remind him/her to turn in funds.

- The head cashier (the person responsible for the deposit control associated with the various tender controls) approves the turn in using the [Deposit Control – Turn Ins](#) page.

All turn-ins must be approved for balancing to complete. A tender control cannot be balanced until the head cashier has approved all turn-in events.

Balancing By Tender Type

It should be noted that when it's time to balance a tender control, the cashier must enter the amount of each tender type that is in the drawer in order to balance it. For example, assume the following takes place:

- A drawer is opened with a starting balance of \$150.50 (tender type is Cash).
- During the day, the cashier receives \$5,000 in Cash, and \$1,000 in Checks.
- During the day, the cashier turns in \$750 of the Checks and \$4,000 of Cash.

At the end of the day, the following operator would have to enter the balances shown in the Ending Balance column.

Tender Type	Starting Balance	Tenders Received	Turn Ins	Ending Balance
Cash	\$150.50	\$5,000	\$4,000	\$1150.50
Check	-	\$1,000	\$750	\$250

Time saver. The system assists in the balancing effort by amalgamating the amount of tenders by tender type when the tender control's status is changed from *Open* to *Balancing In Progress*. The cashier just needs to enter the Ending Balance. The system can then compare the cashier's Ending Balance against the Expected Ending Balance. When these values are equal for all tender types, the tender control's status can become *Balanced*.

Cash Back

When a [payment is added](#), the user defines the following:

- The amount of debt to be relieved (i.e., the payment amount)
- The amount remitted (i.e., the tender amount)
- The form of the remittance (i.e., the [tender type](#))

The payment amount typically equals the tender amount unless cash will be returned to the customer. For example, if a customer remits \$100, but only wants to pay off \$25 of debt, the tender amount will be \$100 and the payment amount will be \$25. The system will only allow a user to remit more than the payment amount if the tender type indicates "cash back" is allowed. For example, you may not allow cash to be returned if a check is remitted, but you may allow it to be returned if cash is remitted.

If cash back is allowed for the tender type, the system displays the amount of cash to be returned on the [Payment Event](#). In addition, because the system enforces [balancing the cash drawer by tender type](#), the system adjusts the payment event's tendered information as follows:

- When there is cash back, the payment event will have two tenders - one will be for the amount and type entered by the user, the other will be a negative amount with a tender type of cash (note, this tender type is retrieved from the Starting Balance Tender Type on the [installation record](#)). For example, if a customer remits \$100 in traveler's checks, but only wants to pay off \$25 of debt, there will be two tenders: one for the \$100 travelers check and the other for the -\$25 of cash.

Alternate currency payments. If your organization accepts payments in alternate currencies, that is, a currency other than the account's currency; the cash back tender is always created in the account's currency.

Multiple tenders and payment cancellation. If multiple tenders were created because of "cash back" processing, both tenders must be cancelled if the payment event needs to be cancelled.

When modifying an unfrozen payment on the [Payment Event](#), if the payment becomes unbalanced, a button is displayed allowing the user to **Recalculate Cash Back**. If the user clicks this button, the system reassesses the cash back tender as follows:

- If there is now cash back, a new tender is created for the credit amount
- If the cash back amount has changed, the tender for the cash back is adjusted to the new amount
- If there is no longer cash back due, the tender for the cash back is removed.

Managing Payments Interfaced From External Sources

Just like a payment recorded on-line via a cash drawer, a payment interfaced from an external source (e.g., lock box or remittance processor) must reference a tender control and the tender controls, in turn, must reference a deposit control. The only real differences between these two types of payments are highlighted below:

- Whilst an operator must create and balance the tender and deposit controls for real-time payments, the system creates the tender and deposit controls associated with interfaced payments.
- It's impossible for an invalid account to be referenced on a payment recorded real-time. However, it is quite possible for an interfaced payment to reference an unknown account. If an invalid account is referenced on an interfaced payment (using no distribution rules), the system links it to the suspense service agreement referenced on the tender source control table.

Refer to [Interfacing Payments From External Sources](#) for more information.

Exceptions

The topics in this section describe exceptions that are detected when the system allocates a payment.

Contents

[Payment Exceptions](#)
[Payment Event Exceptions](#)
[Resolving Exceptions Automatically](#)

Payment Exceptions

When the system attempts to distribute a payment, there are a small number of situations where it can't do its job. Some examples of classic errors:

- **No service agreement to hold a credit.** For example, if an account overpays their debt and the account doesn't have a single service agreement that is allowed to hold a credit, a payment error is generated.

The system saves payments that are in error just as it saves payments that are error-free. This is done because payments are nothing more than a snapshot of the data that was used to distribute the payment. By saving the snapshot, you can see the information the system used when it detected the error and therefore more effectively correct it.

Every payment in error is written to the [Payment Exception](#) table. A To Do background process creates To Do entries for records in this table.

Payment Event Exceptions

It is possible for a payment event's tenders to not equal its payments. Such events are classified as **unbalanced**. Refer to [Unbalanced Payment Events](#) for how this can happen.

For each unbalanced payment event, a record is written to the [Payment Event Exception](#) table. A To Do background process creates To Do entries for records in this table.

Resolving Exceptions Automatically

Some payment errors occur because master data was not fully set up prior to receiving a payment for the account. For these cases, the system will periodically check to see whether the master data problem has been resolved by attempting to distribute and freeze the payment in error.

A background process, Resolve Payments in Error – [PY-RPE](#), exists for this purpose. This background process works as follows:

- It looks for payments in error where the error was caused by the lack of active service agreements linked to the payment's account. Note, this may occur when a customer is verbally asked to pay a connection fee or deposit prior to starting service, and for some reason, the appropriate service agreement has not been set up yet, or is still pending.
- For each such payment, it attempts to re-distribute the payment. If service agreements have been created in the meantime, this payment will distribute and freeze successfully.

Payment Financial Transaction Considerations

A payment segment exists for each service agreement that receives a portion of a payment. Linked to every frozen payment segment is a financial transaction. This financial transaction affects a service agreement's payoff balance and/or current balance. It also contains the journal details that debit cash and credit some other account.

The topics in this section provide information about the financial impact of a payment segment.

Refer to [The Financial Big Picture](#) for more information about a payment's place in the financial big picture.

Contents

[Payment - Current Balance versus Payoff Balance](#)

[The Source Of GL Accounts On A Payment Financial Transaction](#)

Payment - Current Balance versus Payoff Balance

Warning! If you do not understand the difference between payoff balance and current balance, refer to [Current Amount versus Payoff Amount](#).

A payment segment financial transaction almost always affects payoff balance and current balance by the same amount (think of it like this - when a customer pays, the amount they think they owe goes down by the amount they really owe). The only exception is a payment segment for a charitable contribution. These payment segments only affect current balance because the customer was never billed for the contribution in the first place.

Refer to [Setting Up Payment Segment Types](#) for more information about how payment segment type affects how a payment segment is produced and how its financial transaction is generated.

The Source Of GL Accounts On A Payment Financial Transaction

A payment segment's financial transaction also contains the double-sided accounting entry that defines how the payment segment affects the general ledger.

Refer to [The Source Of GL Accounts On Financial Transactions](#) for a description of where the system extracts the distribution codes used to construct the GL accounts.

A Payment May Affect More Than Just Customer Balances

The topics in this section provide information about obscure things that may happen when a payment is distributed and frozen.

Contents

- [Open Item Accounting and Match Events](#)
- [FT Freeze Repercussions](#)

Open Item Accounting and Match Events

Refer to [Payments and Match Events](#) for more information about how payments can create match events for open-item accounts.

FT Freeze Repercussions

Refer to [Obscure Things That Can Happen](#) for more information about things that can happen when an FT is frozen (and FT's get frozen when a payment is frozen).

Automatic Payments

This section discusses how to set up and manage customers who pay their bills automatically (via direct debit or credit card debits)

Contents

- [How To Set Up A Customer To Pay Automatically](#)
- [What Are Automatic Payments?](#)
- [How And When Are Automatic Payments Created?](#)
- [Automatic Payment Dates](#)
- [How To Implement Maximum Withdrawal Limits](#)
- [How Are Automatic Payments Cancelled?](#)
- [Match Events Are Created For Open-Item Customers When An Automatic Payment Is Created](#)
- [Pay Plans and Automatic Payments](#)
- [Non-billed Budgets and Automatic Payment](#)
- [Downloading Automatic Payments and Interfacing Them To The GL](#)
- [ACH Record Layouts](#)

How To Set Up A Customer To Pay Automatically

If a customer wants to pay automatically, transfer to [Account – Auto Pay](#) and define the source of the funds and the customer's account or credit card number.

What Are Automatic Payments?

An automatic payment is just like any other payment (refer to [A Payment Event Has Payments And Tenders](#) for more information about payments in general). However, automatic payments have one special trait – they cause funds to be transferred into your company's bank account. Refer to [Downloading Automatic Payments and Interfacing Them To The GL](#) for how this transference happens.

How And When Are Automatic Payments Created?

Automatic payments can be created in several ways:

- The system creates automatic payments for bills linked to accounts with an active auto pay option (note, the system uses the [account's autopay option](#) that's effective on the bill's due date). When the system does this is dependent on the value of the **Autopay Creation Option** on the [installation record](#):
 - If you've set this option to **Create At Bill Completion**, the automatic payment is created when the account's bill is **completed**. The payment is distributed and frozen at the same time. This means that the account's balance is almost always zero (or in credit if they have a deposit). You will see the automatic payments in the payment list immediately after the bill is completed (rather than waiting until the automatic payment's extraction date).
 - If you've set this option to **Create On Extract Date**, the automatic payment is NOT created when the bill is **completed**. Rather,
 - At bill completion time, the bill is stamped with the automatic payment's extract date and amount. The date is the automatic payment source's extraction date (refer to [Automatic Payment Dates](#) for more information on how this date is calculated).
 - The automatic payment background process ([APAYCRET](#)) creates the automatic payment on the extract date stamped on the bill.
 - The automatic payment is NOT distributed and frozen when the automatic payment is initially created. A separate background process ([APAYDSFR](#)) distributes and freezes the automatic payment on the automatic payment GL distribution date (refer to [Automatic Payment Dates](#) for more information on how this date is calculated). This means that the customer's balance increases when the bill is completed and is only reduced when the automatic payment is marked for interface to the general ledger.

Note that it is possible for automatic payments to be distributed and frozen after being extracted and interfaced to a financial institution. Please refer to [Downloading Automatic Payments and Interfacing Them To The GL](#) and [The Nightly Processes](#).

An algorithm plugged in on the [Installation Record](#) calculates the payment amount whether the automatic payment is created at bill completion time or on the extract date. Please refer to [APAM-DFLT](#) for more information about how the algorithm that is supplied with the base package calculates this amount.

With balance forward accounting, automatic payments are not just for new charges. The base package algorithm includes prior balances when it creates a customer's first automatic payment. For example, if a customer has an existing balance of \$100 and then signs up for automatic payment, their next bill will cause an automatic payment of \$100 plus any new charges to be created (assuming the \$100 remains unpaid at the time the next bill is completed). Refer to [Open Item Versus Balance Forward Accounting](#) for information about balance forward accounting.

- If a customer with a pay plan indicates that the payment method is via “autopay”, a background process ([PPAPAY](#)) creates an automatic payment on the scheduled payment dates. Please note, if the **Autopay Creation Option** on the [installation record](#) is set to **Create On Extract Date**, the automatic payment is NOT distributed and frozen when the automatic payment is initially created. Rather, a separate background process ([APAYDSFR](#)) distributes and freezes the automatic payment on the automatic payment GL distribution date (refer to [Automatic Payment Dates](#) for more information on how this date is calculated). Refer to [The Big Picture Of Pay Plans](#) for more information about pay plans.
- If a customer with an account that is set up for automatic payment has a non-billed budget that is not excluded from automatic payment, a background process ([NBBAPAY](#)) creates an automatic payment on the scheduled payment dates. Please note, if the **Autopay Creation Option** on the [installation record](#) is set to **Create On Extract Date**, the automatic payment is NOT distributed and frozen when the automatic payment is initially created. Rather, a separate background process ([APAYDSFR](#)) distributes and freezes the automatic payment on the automatic payment GL distribution date (refer to [Automatic Payment Dates](#) for more information on how this date is calculated). Refer to [What is a Non-billed Budget](#) for more information about non-billed budgets.
- A user can create an automatic payment by simply adding a payment tender with a tender type that indicates it is for automatic payment purposes. This would be a rather unusual thing to do, but you might do this if you want to immediately debit a customer's bank account after a large adjustment is added to the system (e.g., if they suddenly owe you a lot of money and you don't want to wait until the next bill to collect it). Automatic payments created by this method must be distributed and frozen before they can be extracted.

An algorithm is used to create automatic payments. The logic used to create automatic payments is plugged in on the [Installation Record](#). Please note that this algorithm is **not** called when a user manually creates an automatic payment (by adding a payment tender with a tender type that indicates that it is for automatic payment purposes).

When an automatic payment is first created, it gets marked with a distribution date. The distribution date is the date on which the automatic payment's FT's GL details can be interfaced to the general ledger (via the standard GL interface). The distribution date is determined as follows:

- Every automatic payment references an auto-pay source.
- Every auto-pay source references an auto-pay route type.
- Every auto-pay route type contains an algorithm that is responsible for calculating the GL Distribution (Posting) date. On the GL distribution date, the automatic payment will be interfaced to the general ledger.

Automatic Payment Dates

As described in the previous section, an algorithm (that's plugged in on [Auto Pay Route Types](#)) controls the date on which the automatic payment is interfaced to your general ledger. We refer to this date as the GL distribution date.

This algorithm also populates the following dates:

- The payment date that is stored on the payment.
- The date on which the automatic payment is interfaced to the financial institution.

The algorithm that is supplied with the base package provides many parameters that allow you to dictate how these dates are calculated. Please refer to [APAY-DTCALC](#) for the details.

How To Implement Maximum Withdrawal Limits

In some locales, customers can define a "maximum withdrawal amount" to limit the amount of money that is automatically debited from their bank account. For example, a low-income customer may want to prevent direct debits of more than \$50 from being applied to their checking account.

You define a customer's maximum withdrawal amount when you setup their automatic payment information on [Account – Auto Pay](#).

The following points describe how the system implements maximum withdrawal limits:

- When a bill is completed for a customer who pays automatically the system calls the automatic payment creation algorithm that's plugged in on the [Installation Record](#). The [base-package autopay creation algorithm](#) checks if the amount of the automatic payment exceeds the customer's maximum withdrawal amount. If so, the autopay creation algorithm calls the [automatic payment over limit algorithm](#) that's plugged in on the account's [customer class](#). Algorithms of this type have the ability to reduce the amount of the autopay or to prevent the autopay from being created. Refer to [APOL-RA](#) for an example plug-in.
- When a user manually creates an automatic payment (by adding a tender with a tender type that indicates that it is for automatic payment purposes), the system issues a warning message when the tender amount exceeds the account's maximum withdrawal amount.

Pay plans and non-billed budgets. Please note that automatic payments that are created as a result of pay plans and non-billed budgets are not subject to maximum withdrawal limits. This is because both of these options required customer approval and therefore the customer should be able to plan accordingly. Refer to [How And When Are Automatic Payments Created](#) for more information.

How Are Automatic Payments Cancelled?

There are two ways to cancel an automatic payment:

- The system will cancel an automatic payment behind-the-scenes if the related bill (if any) is reopened BEFORE the automatic payment is interfaced to the financial institution. When you recomplete the bill, the system will create a new automatic payment that reflects the new amount due (and the canceled automatic payment will net out the original automatic payment).
- An operator can cancel an automatic payment (refer to [How To Cancel A Tender](#)) at any time. You would do this if the automatic payment was rejected by the financial institution.

Refer to [Credit / Correction Notes and Auto Pay](#) for information about how a credit note may cancel an automatic payment.

Match Events Are Created For Open-Item Customers When An Automatic Payment Is Created

The system creates a match event when a bill is completed for open-item customers that pay automatically (i.e., direct debit customers). The match event groups together the bill's new charges against the automatic payment's payment segments.

If the bill is subsequently re-opened, the match event will be cancelled when the automatic payment is cancelled.

Refer to [Open Item Accounting](#) for more information.

Pay Plans and Automatic Payments

If a customer wants to pay their pay plan scheduled payments automatically, the account must be set up for automatic payment (as described under [How To Set Up A Customer To Pay Automatically](#)). In addition, the payment method defined on the pay plan must indicate automatic payment is being used.

When this is done, a background process referred to as **PPAPAY** creates automatic payments on the scheduled payment date by calling the automatic payment creation algorithm plugged in on the installation record.

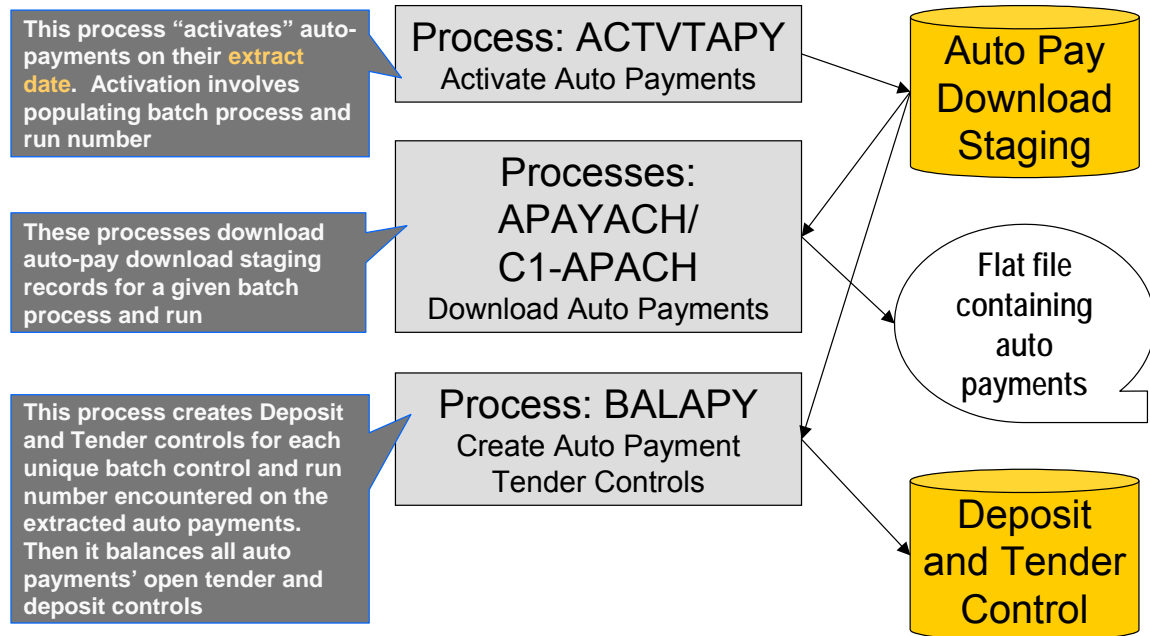
Non-billed Budgets and Automatic Payment

If a customer wants to pay their non-billed budget scheduled payments automatically, the account must be set up for automatic payment (as described under [How To Set Up A Customer To Pay Automatically](#)). In addition, the non-billed budget must indicate that automatic payment is being used.

When this is done, a background process referred to as **NBBAPAY** creates automatic payments on the scheduled payment date by calling the automatic payment creation algorithm plugged in on the installation record.

Downloading Automatic Payments and Interfacing Them To The GL

The following diagram illustrates the background processes that interface automatic payment out of the system:



These processes are described in the following topics.

Contents

[ACTVTAPY - Activating Automatic Payments](#)

[APAYACH - Download Automatic Payments To The ACH \(automated clearing house\)](#)

[C1-APACH - Download Automatic Payments To The ACH \(automated clearing house\)](#)

[BALAPY - Creating Automatic Payment Tender Controls](#)

ACTVTAPY - Activating Automatic Payments

When an automatic payment is first created, it gets marked with an extract date. The extract date is the date the automatic payment will be downloaded to the respective financial institution. The extract date is determined as follows:

- Every automatic payment references an auto-pay source.
- Every auto-pay source references an auto-pay route type.
- Every auto-pay route type contains an algorithm that calculates this date.

On the extract date, the automatic payment is “activated”. The automatic payment is activated is marked for download the next time its download process runs. An automatic payment’s download process is defined on its auto-pay source’s route type.

APAYACH - Download Automatic Payments To The ACH (automated clearing house)

This process reads all auto pay download staging records marked with a given batch control ID & run number and creates the flat file that’s passed to the ACH. Refer to [ACH Record Layouts](#) for the details of the record layouts.

This process may be rerun. 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 a different flat file format, you must create additional versions of this program. Refer to [Setting Up Automatic Payment Extracts](#) for instructions describing how to add another process.

C1-APACH - Download Automatic Payments To The ACH (automated clearing house)

This process is an alternative to using APAYACH. Similar to APAYACH, this process reads all auto pay download staging records marked with a given batch control ID and run number and creates the flat file that's passed to the ACH. Refer to [ACH Record Layouts](#) for the details of the record layouts.

The following outlines the differences between APAYACH and C1-APACH:

- C1-APACH populates the Effective Entry Date field on the Company Batch Header Record with a date (the actual draft date) as to when the payment should be withdrawn from the customer's account. APAYACH uses the Business Process Date as the draft date.
- C1-APACH produces a file with multiple occurrences of the Company Batch Header Record differing on the Effective Entry Date field. APAYACH produces a file that has 1 File Header Record, 1 Company Batch Header Record and multiple detail lines.

This process may be rerun. You can reproduce the flat file at any time. Request this job and specify the run number associated with the historic run.

If you require a different flat file format, you must create additional versions of this program. Refer to [Setting Up Automatic Payment Extracts](#) for instructions describing how to add another process.

BALAPY - Creating Automatic Payment Tender Controls

This process creates a new tender control (with an associated deposit control) for each unique batch control and run number encountered in the extracted automatic payments (where its payment tender is not yet linked to a tender control). The payment tender of each of these automatic payments is then linked to the corresponding tender control. This process also balances the open tender control records afterwards.

ACH Record Layouts

The topics in this section describe the layout of the records created by [APAYACH – Download Automatic Payments To The ACH \(automated clearing house\)](#) and [C1-APACH - Download Automatic Payments To The ACH \(automated clearing house\)](#).

Note. APAYACH and C1-APACH share most of the same fields. The exceptions are documented in the following tables.

Contents

- [File Header Record](#)
- [Company Batch Header Record](#)
- [Entry Detail Record](#)
- [Company Batch Control Record](#)
- [File Control Record](#)

File Header Record

The ACH extract flat file must have one record of this type and it must be the first logical record on file.

Field Name	Format	Source/Value/Description
RECORD-TYPE	A1	"1"
PRIORITY-CD	A2	"01"
RESERVED-01	A1	Spaces
IMMEDIATE-DESTINATION	A9	CI_BANK_ACCOUNT.DFI_ID_NUM
COMPANY-ID	A10	CI_BANK_ACCOUNT.ACCOUNT_NBR
FILE-CRE-DT	A6	YYMMDD. Current date
FILE-CRE-TM	A4	HHMM. Current time
FILE-ID-MODIFIER	A1	"A"
RECORD-SIZE-CONST	N3	"094" The "FILLER" at the end of all but the Entry Detail Record serves to bring the total length of each record up to this constant.
BLOCKING-FACTOR	N2	"10"
FORMAT-CD	A1	"1"
ORIG-FIN-INST-NAME	A23	CI_BANK_L.DESCR
COMPANY-NAME	A23	CI_BANK_ACCOUNT_L.DESCR
REFERENCE-CD	A8	Spaces

Company Batch Header Record

The ACH extract flat file must have one record of this type and it must be the second logical record on file.

Field Name	Format	Source/Value/Description
RECORD-TYPE	A1	"5"
SERVICE-CLASS-CD	A3	"200"
COMPANY-NAME	A16	CI_BANK_ACCOUNT_L.DESCR
COMPANY-DISCRETIONARY	A20	Spaces
COMPANY-ID	A10	CI_BANK_ACCOUNT.ACCOUNT_NBR
STD-ENTRY-CLASS	A3	"PPD"
CO-ENTRY-DESCR	A10	"PAYMENT"
EFF-ENTRY-DT Note. This field applies only to C1-APACH.	A6	CI_APAY_CLR_STG.SCHED_EXTRACT_DT + NBR-DAYS batch parameter if specified greater than 0
COMPANY-DESCR-DT	A6	Business process date
EFF-ENTRY-DT	A6	Business process date
RESERVED-01	A3	Spaces
ORIGINATOR-STAT-CD	A1	"1"
ORIGIN-DFI-ID	A8	CI_BANK_ACCOUNT.DFI_ID_NUM

Field Name	Format	Source/Value/Description
BATCH-NBR	N7	The batch number of this batch within the file.

Entry Detail Record

The ACH extract flat file must have one record of this type for every direct debit record.

Field Name	Format	Source/Value/Description
RECORD-TYPE	A1	"6"
TRANSACTION-CD	A2	CI_TENDER_TYPE.EXT_TYPE_FLG
TRANSIT-RTG-NBR	A9	CI_APAY_SRC.EXT_SOURCE_ID
DFI-ACCT-NBR	A17	CI_APAY_CLR_STG.EXT_ACCT_ID
AMOUNT	N8.2	CI_PAY_TNDR.TENDER_AMT
INDIVIDUAL-ID	A15	CI_PAY_TNDR.PAYOR_ACCT_ID
INDIVIDUAL-NAME	A22	CI_APAY_CLR_STG.ENTITY_NAME
DISCRETIONARY-DATA	A2	Spaces
ADDENDA-REC-IND	A1	"0"
DFI-ID	A8	CI_BANK_ACCOUNT.DFI_ID_NUM
ENTRY-DETAIL-CNT	N7	Ascending counter starting at 1 and incremented by 1 for each entry detail record

External Account ID. The EXT_ACCT_ID field supports up to 50 characters. If the value entered in the field is longer than that supported by the record layout (17 characters), the value will be truncated.

Company Batch Control Record

The ACH extract flat file must have one record of this type and it must follow the Entry Detail records.

Field Name	Format	Source/Value/Description
RECORD-TYPE	A1	"8"
SERVICE-CLASS-CD	A3	"200"
ENTRY-ADDENDA-CNT	N6	The total number of entry detail records in this batch.
ENTRY-HASH	N10	The product of the first 8 digits of the external source id of the autopay source, multiplied by the number of entry detail records in the batch.
TOTAL-DR-DOLLAR-AMT	N10.2	Total tender amounts of the entry detail records.
TOTAL-CR-DOLLAR-AMT	N10.2	Zero.
COMPANY-ID	A10	CI_BANK_ACCOUNT.ACCOUNT_NBR
RESERVED-01	A19	Spaces
RESERVED-02	A6	Spaces
ORIGIN-DFI-ID	A8	CI_BANK_ACCOUNT.DFI_ID_NUM
BATCH-NBR	N7	The batch number of this batch within the file.

File Control Record

The ACH extract flat file must have one record of this type and it must be the last logical record on file.

Field Name	Format	Source/Value/Description
RECORD-TYPE	A1	"9"
BATCH-CNT	N6	The number of batches in this file.
BLOCK-CNT	N6	Calculation of the total number of records in the file / 10.0 + 0.9
ENTRY-ADDENDA-CNT	N8	The total number of entry detail records in this file.
ENTRY-HASH	N10	The sum of the entry hash values on all the batch control records in this file.
TOTAL-DR-DOLLAR-AMT	N10.2	Sum of the total debit entry dollar amounts of all the batches in this file.
TOTAL-CR-DOLLAR-AMT	N10.2	Sum of the total credit entry dollar amounts of all the batches in this file.
RESERVED-01	A39	Spaces

Issuing A Payment Advice Instead Of Creating An Automatic Payment

If the system is configured to send the customer a payment advice (instead of initiating an electronic funds transfer) when a bill is completed, the automatic payment records – i.e. payment event, payment, tender and auto pay clearinghouse staging – are not created. Refer to [Payment Advices](#) for more information.

Contents

[How To Set Up A Customer To Receive Payment Advices](#)
[Payment Advice Option Is For Bill-Related Automatic Payments Only](#)

How To Set Up A Customer To Receive Payment Advices

Use [Account – Auto Pay](#) to capture the customer's bank details and indicate an auto pay method of **Payment Advice**.

Payment Advice Option Is For Bill-Related Automatic Payments Only

Payment advices can be printed for auto pays that result from completed bills.

An auto pay for a pay plan scheduled payment will not be created if the account's effective auto pay option is set to **Payment Advice**. The Pay Plan Auto Pay (**PPAPAY**) batch process will log an error in this case.

An auto pay for a non-billed budget scheduled payment will not be created if the account's effective auto pay option is set to **Payment Advice**. The Non-Billed Budget Pay Plan Auto Pay (**NBBAPAY**) batch process will log an error in this case.

Manually created automatic payments (i.e. auto pays created via payment event UI) are always processed as direct debit.

Maintaining Payment Events

A payment event is used to record when moneys are remitted and how the moneys are allocated amongst accounts. The topics in this section describe how to maintain payment events.

The system creates most payment events behind-the-scenes. Most payment events are created by the system when it [uploads payments](#) and when it creates [automatic payments](#). You should only have to access the payment event transaction if you need to correct a payment event or add a payment event real-time. For information about how the system creates payment events, refer to [The Big Picture of Payments](#).

Contents

- [Payment Lifecycles](#)
- [Payment Event - Add Dialog](#)
- [Payment Event - Main Information](#)
- [Payment Event - Tenders](#)
- [Payment Event - Characteristics](#)
- [Payment Event Action Codes](#)

Payment Lifecycles

The topics in this section describe the lifecycle of the various payment objects.

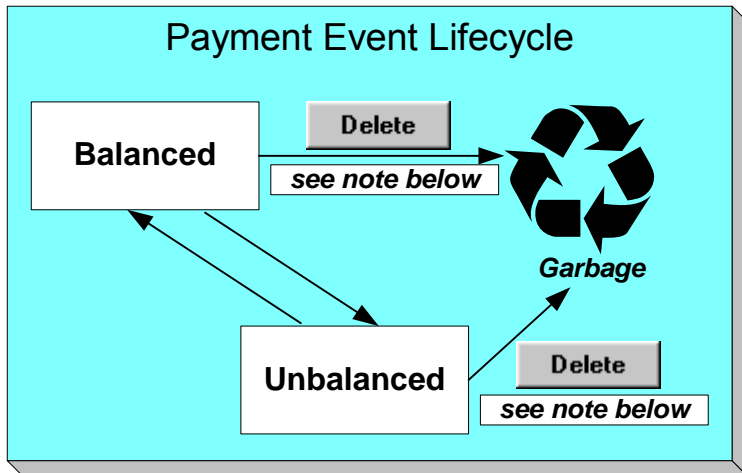
Contents

- [Payment Event Lifecycle](#)
- [Tender Lifecycle](#)
- [Payment Lifecycle](#)

Payment Event Lifecycle

The following diagram shows the possible lifecycle of a payment event.

Warning! This diagram only makes sense in the context of the page used to maintain payment events. Refer to [Payment Event - Main Information](#) for the details.



The system, by default, distributes the sum of a payment event's tenders to the account that remits the tenders. After distribution the sum of the tenders equals the sum of the payments (remember, the term *payment* is used to refer to an allocation of some/all of a payment event's tenders to an account's debt) when the event is first created. We refer to such an event as being **Balanced**.

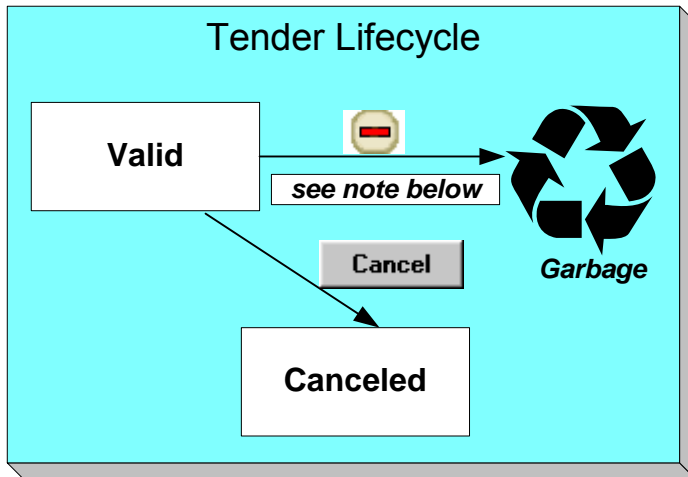
It is possible via any of the methods described in [Unbalanced Payment Events](#) to make a balanced payment event **Unbalanced**.

Click **Delete** to physically remove a balanced or unbalanced payment event from the database. You may not delete a payment event if: a) there are frozen or canceled payments linked to the event, or b) if there are canceled tenders linked to the event, or c) if a tender linked to the event is part of a balanced tender control. When the payment event is deleted, the system also deletes its tenders, payments, and payment segments.

Tender Lifecycle

The following diagram shows the possible lifecycle of a payment event.

Warning! This diagram only makes sense in the context of the page used to maintain tenders. Refer to [Payment Event - Tenders](#) for the details.



A tender is initially saved in the **Valid** state. It is possible via any of the methods described in [Unbalanced Payment Events](#) to make a balanced payment event **Unbalanced**.

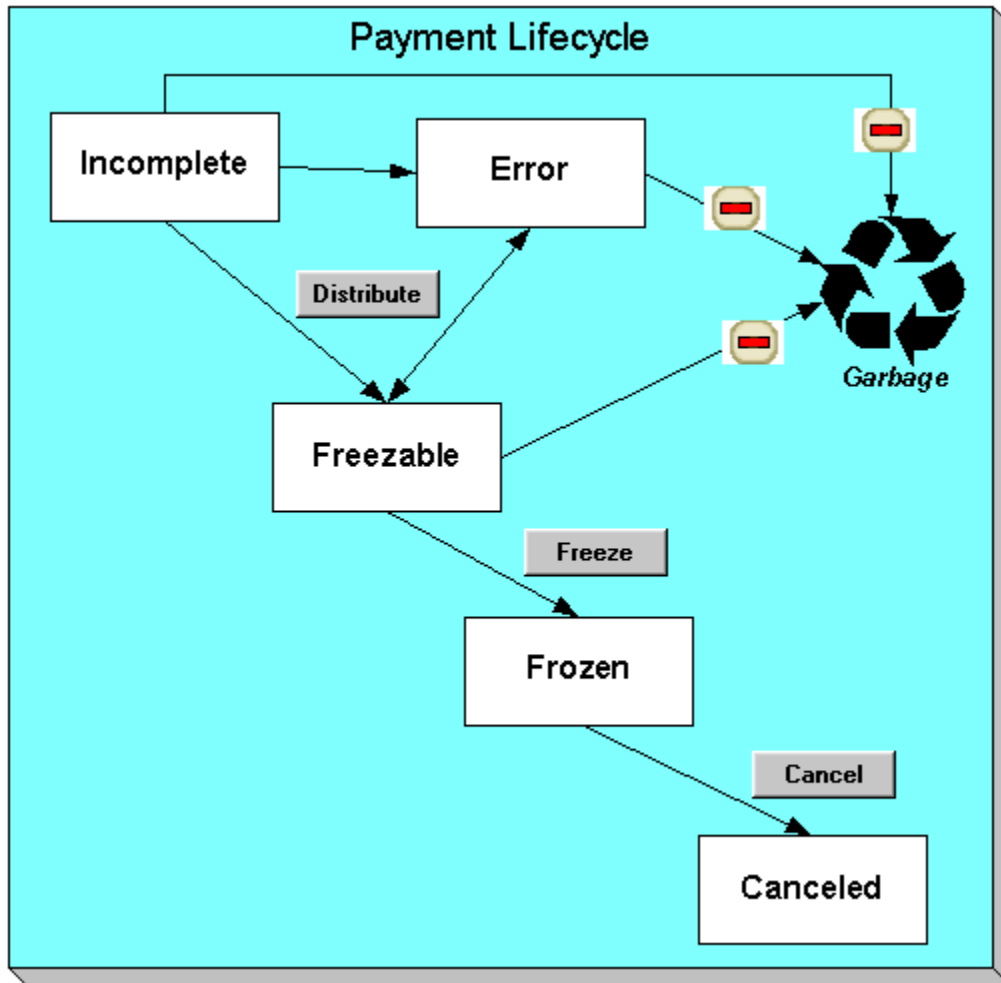
If a tender is invalid, click **Cancel** to cancel the tender AND ALL PAYMENTS LINKED TO THE EVENT.

Click the delete button to physically remove a tender from the database. You may not delete a tender if it is part of a balanced tender control.

Payment Lifecycle

The following diagram shows the possible lifecycle of a payment.

Warning! This diagram only makes sense in the context of the page used to maintain a payment. Refer to [Payment - Main](#) for the details.



Payments are initially created in the **Incomplete** state. Payments in this state don't have payment segments or financial transactions; they are simply a stub awaiting distribution.

Click **Distribute** to distribute a payment amongst an account's service agreements.

- If the system cannot distribute the payment (for whatever reason), the payment is moved to the **Error** state. You may delete such a payment.
- If the system successfully distributes a payment, the payment becomes **Freezable**.

Click the - button to physically remove an **Incomplete**, **Error** or **Freezable** payment from the database.

Click **Freeze** to freeze a payment and its financial transaction. Freezing the payment causes the following to occur:

- The system executes any payment freeze algorithms linked to the account's [customer class](#) and to the service agreement's [SA type](#).
- The payment's state becomes **Frozen** and the payment may now appear on a customer's bill.

You may not change a payment once it is frozen. However, you may reverse the payment's financial effect by clicking **Cancel**. Clicking this button will cause the following to occur:

- A new financial transaction is generated and linked to the payment. This financial transaction reverses the financial effects of the original payment.
- The system executes any payment cancellation algorithms linked to the account's [customer class](#).
- If the payment has a related adjustment, this adjustment is also cancelled. Refer to [Loan Overpayments](#) for an example of payments with adjustments.
- The payment becomes **Canceled**.

Payment Event - Add Dialog

The Payment Event transaction features an unusual dialog that simplifies the addition of new payment events. This page appears if you open the **Financial, Payment Event** page in add mode from either the Account context menu or from the main menu (it also appears if you click the clear button when on the Payment Event page).

Note. If you have opted to always use the payment event distribution rules method as your [default method](#), the [Payment Event Quick Add \(Single Payment Event\)](#) page appears instead.

Description of Page

The **Payor Account ID** is the account that remitted the payment. We assume this account is both the tendering account and the account whose debt is being relieved by the payment. If this assumption is not correct, choose a **Distribute Action** of **Do Not Distribute** and then change the tendering or paying account when the Payment Event page appears (after you click **OK**).

Default note. If you have navigated to this page from account context menu, the Payor Account ID and Payment Amount are defaulted to this account.

The **Payment Amount** is the amount of the customer's debt to be relieved by the payment. Note, this amount is defaulted using an algorithm plugged in on the [Installation Record](#). Please refer to [APAM-DFLT](#) for more information about how the algorithm that is supplied with the base package calculates this amount.

The payment tenders grid allows you to enter multiple tender types and amounts. Click + to add a new tender. For each tender specify the following fields:

- The **Amount Tendered** is the amount of moneys remitted for the tender type.
- **Tender Type** describes what was remitted (e.g., cash, check, ...). **Tender Type** defaults from the Quick Add Tender Type that is defined on the [installation record](#).
- If a check was tendered, use **Check Number** to specify the identity of the check.

Cash back causes an additional tender to be created. If cash should be returned to the customer (because the customer overpaid and the tender type's cash back allowed switch is true and the tender type is not "like cash"), a negative tender for the cash back amount is created for the payment event. Refer to [Cash Back](#) for a description of how the system can recommend Cash Back amount if the customer tendered more than they are paying.

Match Type and **Match Value** are used if either of the following conditions is true:

- This **Payor Account** belongs to an [open item](#) customer class. In this situation, specify a **Match Type** to define how the payment should be matched to the customer's open-items and use **Match Value** to define the open-items covered by the payment. For example, if this payment is in respect of a bill, specify a match type of "bill id" and a match value of the bill id being paid.

Shortcut. If you enter a **Match Type** of "bill id" and leave the **Match Value** blank, the system assumes the customer wants to pay the latest bill.

- The customer wants to restrict the distribution of the payment to a specific service agreement. In this situation, specify a **Match Type** of "service agreement ID" and a **Match Value** of the respective service agreement ID.

The **Payment Date** defaults to the current date.

If the **Payor Account ID's** [customer class](#) is designated as non-CIS (i.e., the person making the payment isn't a customer), the following information appears in the above window:

- **Non CIS Name** is the name of the person remitting the payment.
- **Reference Number** is the reference number of the item being paid (e.g., the property tax reference number).
- **Non CIS Comments** are used to describe anything unusual about the non-CIS payment.

Use **Distribute Action** to describe what you'd like to have happen when you click the **OK** button:

- Choose **Distribute and Freeze if OK** if this is a simple payment that should require no manual intervention. By "simple payment" we mean:
 - The account is both the tendering account and the account whose debt is being relieved by the payment
 - The payment date is the current date
 - The payment should be [distributed](#) amongst the account's service agreements using standard distribution logic

If this option is selected, the system distributes the **Payment Amount** amongst the account's service agreements. If the distribution is successful, the system automatically freezes the payment. If the distribution is not successful, the payment will be in the **Error** or **Incomplete** state. When the Payment Event page appears, you can view the error and then correct it. After the cause of the error is corrected, you must distribute and freeze the payment manually (this can be done on several pages including [Payment Event - Main](#) and [Payment - Main](#)).

- Choose **Manual Distribution** if you need to manually distribute the payment amongst the account's service agreements. If this option is selected, the system creates a payment event, a tender and a payment and then transfers you to the [Payment - Manual Distribution](#) page where you can define the amount to be allocated to each of the account's service agreements. After you've distributed the payment, don't forget to freeze it.
- Choose **Do Not Distribute** if you want to process the payment event manually (e.g., if you need to define multiple accounts whose debt is relieved by the payment). If this option is selected, the [Payment Event – Main](#) page opens with the information you entered defaulted accordingly. You can make any changes you want and then distribute and freeze the payment. Refer to [How To Add A New Payment Event](#) for more information.

Cash only warning. If the account has exceeded your [cash-only threshold](#) and the **Tender Type** isn't marked as "like cash", a warning appears advising of such. Note, a customer's cash-only points are maintained on [Account - Credit Rating](#).

Payment Event - Main Information

The Main page contains core payment event information. Open this page using **Financial, Payment Event**.

The **Description of Page** section below describes the fields on this page. Refer to [How To](#) for a description of how to perform common payment event maintenance functions.

Description of Page

Cash-only customers. When you attempt to add a new payment event, the system warns you if the account remitting the tender is a cash-only customer and the tender's [tender type](#) is not cash. An account is considered cash-only if their current cash-only points exceed the cash-only tolerance maintained on the CIS Installation record. A customer's cash-only points are maintained on [Account - Credit Rating](#).

Pay Event Info contains a concatenation of the payment date, amount, and the name of the main customer on the account that remits the tender. If multiple tenders exist, the customer's name is not displayed. If the payment event is associated with a single distribution detail, the rule name and the description of the rule value are displayed as well. If multiple distribution details exist, **Multiple Distribution Details Exist** is displayed instead. If document numbering has been enabled, the assigned document number will appear. If the payment was made in an alternate currency, the message **Alternate Currency Used** is displayed. **Pay Event Info** is only displayed after the payment event has been added to the database.

Refer to [Alternate Currency Payments](#) and [Document Numbers](#) for more information. **Payment Event ID** is the system-assigned unique identifier of the payment event.

The area under **Pay Event Info** provides warnings about the payment event. Possible warnings are **Unfrozen Payments** and **The Payment Event is Unbalanced**. A warning is also issued to the cashier to remind him/her to turn in funds – see [Turn Ins](#).

If multiple distribution details are linked to the payment event, the distribution rule, value, and amount for each distribution detail displays.

Payment Date is the business date associated with the payment event.

Warning! If you change the payment date and this event's tender(s) are automatic payments, the extract date (i.e., the date the automatic payment is sent to the financial institution) will be changed to equal the payment date. Refer to [Automatic Payments](#) for more information.

Payment(s) contains the total number and value of payments linked to the event.

Tender(s) contains the total number and value of tenders linked to the event.

Amount Tendered contains the amount that was tendered by the customer.

The system displays a **Cash Back Amount** if the customer tenders more than they are paying (refer to [Cash Back](#) for details).

If the payment event becomes **unbalanced**, a **Recalculate Cash Back** button appears. If this button is clicked, the system creates, deletes or recalculates the cash back tender. Refer to [Cash Back](#) for details.

While most payment events contain a single payment, the system allows many payments to exist under a payment event (when the payment event's tender(s) are distributed amongst multiple accounts). If a payment event has a large number of payments, you can use the **Account Filter** to limit the payments that appear in the **Payments** grid. The following options are available:

- **Account.** Use this option if you only want to see the payment linked to an **Account ID**.
- **All.** Use this option to view all payments linked to the payment event.
- **Person Name.** Use this option to restrict payments linked to accounts whose main customer has a primary name that matches **Person Name**.

The **Payments** grid contains the payment(s) linked to the payment event. The following points describe the attributes in this scroll; refer to [How To](#) for instructions describing how to perform common maintenance activities.

- **Account ID** references the payment's account. The name of the account's main customer is displayed adjacent.
 - **Payment Amount** is the amount of the account's debt relieved by the payment. The adjacent context menu allows you to drill down to the details of the [payment](#) (this is where you can see the payment's payment segments and where you can override the distribution of the payment).
 - **Payment Status** is the payment's status. If the payment's status is **Error**, the error message is displayed adjacent. Refer to [Payment Lifecycle](#) for the potential values and how to handle a payment when it exists in a given state.
 - **Match Type** and **Match Value** should only be used if either of the following conditions is true:
 - This **Account ID** belongs to an [open item](#) customer class. In this situation, specify a **Match Type** to define how the payment should be matched to the customer's open-items and use **Match Value** to define the open-items covered by the payment. For example, if this payment is in respect of a bill, specify a match type of "bill id" and a match value of the bill id being paid.
 - The customer wants to restrict the distribution of the payment to a specific service agreement. In this situation, specify a **Match Type** of "service agreement ID" and a **Match Value** of the respective service agreement ID.
 - The remaining columns are only used if the payment is linked to an **Account ID** that belongs to a customer class that is used for non-CIS payments. Refer to [Setting Up Customer Classes](#) for more information. If such an account exists, the following fields must be defined.
 - **Non CIS Name** is the name of the person remitting the payment.
 - **Reference Number** is the reference number of the item being paid (e.g., the property tax reference number).
 - **Non CIS Comments** are used to describe anything unusual about the non-CIS payment.
- Note, you can also define a comment for a non-CIS payment. To define a comment, use the context menu adjacent to **Payment Amount** to drill to [Payment - Main](#).

- **Payment ID** is the unique, system-assigned identifier of the payment. This value only appears after the payment has been added to the database.

Refer to [Payment Actions](#) and [Payment Event Actions](#) for information about the action buttons on this page. Refer to [How To](#) for a description of typical business processes that use these buttons.

Payment Event - Tenders

The Tenders page contains a scroll showing one row for every tender associated with the payment event. Open this page using **Financial, Payment Event, Tenders**.

The **Description of Page** section below describes the fields on this page. Refer to [How To](#) for instructions describing how to perform common maintenance functions.

Description of Page

Pay Event Info contains a concatenation of the payment date, amount, and the name of the main customer on the account that remits the tender. If multiple tenders exist, the customer's name is not displayed. If the payment event is associated with a single distribution detail, the rule name and the description of the rule value are displayed as well. If multiple distribution details exist, **Multiple Distribution Details Exist** is displayed instead. If document numbering has been enabled, the assigned document number will appear. If the payment was made in an alternate currency, the message **Alternate Currency Used** is displayed. **Pay Event Info** is only displayed after the payment event has been added to the database.

Payment Event ID is the system-assigned unique identifier of the payment event.

Payment Date is the business date associated with the payment event.

Payment(s) contains the total number and value of payments linked to the event.

Tender(s) contains the total number and value of tenders linked to the event.

Amount Tendered contains the amount that was tendered by the customer.

The system displays a **Cash Back Amount** if the customer tenders more than they are paying (refer to [Cash Back](#) for details).

If the payment event becomes **unbalanced**, a **Recalculate Cash Back** button appears. If this button is clicked, the system creates, deletes or recalculates the cash back tender. Refer to [Cash Back](#) for details.

The **Tenders** scroll controls the display of the tenders linked to the payment event. The following simply describes the fields in this scroll; refer to [How To](#) for instructions describing how to perform common tender maintenance activities.

- **Payor Account ID** references the tendering account. The name of the account's main customer is displayed adjacent. If the account has exceeded your cash-only threshold, the warning **Cash only** appears.

Warning! If you change the Payor Account ID and the tender has an associated automatic payment request, the automatic payment request will be removed. A new automatic payment request will be created for the new Payor Account ID. Refer to [Automatic Payments](#) for more information.

- **Tender Amount** is the amount of the tender. The amount is disabled if the payment was made in an alternate currency.
- If a check was used, the **Check Number** contains the identity of the check.
- **Pay Tender ID** is the system-assigned unique identifier of the tender. This value appears after the tender has been added to the database.
- **Tender Type** describes what was remitted (e.g., cash, check). If the **Tender Type** is associated with an automatic payment, the page displays a section that includes information about how and when the automatic payment was interfaced to the payment source.

Authorizing Tenders. The **Tender Type** dropdown list does not include tender types that require authorization (i.e. credit card payments).

For information about tender types requiring authorization, refer to [Credit Card Payments](#).

- If the payment tender was remitted in an alternate currency, i.e. a currency other than the payor account's currency, **Alternate Currency** and **Exchange Rate** will display the formatted alternate amount ((includes the alternate amount and currency) and the exchange rate that was in effect on the payment date.

For information about accepting payments in alternate currencies, refer to [Alternate Currency Payments](#).

- **Tender Status** is the tender's status. Refer to [Tender Lifecycle](#) for the potential values and how to handle a tender when it exists in a given state.
- If the **Tender Type** is associated with an [automatic payment](#), the system attempts to default automatic payment information from the [account's auto-pay](#) option if the tender type is the same as the tender type on the account's auto-pay source and if the auto pay option is effective on the payment date. If the system is unable to default information, you must specify the source of the funds and the customer's account number / credit card number at the financial institution. You can override automatic payment information as long as the automatic payment has not been sent to the financial institution or cancelled. The **Auto Pay Section** contains the following fields:
 - **Auto Pay Source Code** is the financial institution / credit card company that receives the automatic payment request.
 - **Scheduled Extract Date** is the date the automatic payment request was sent / is scheduled to be sent to the financial institution. This information is display-only.
 - **External Account ID** is the customer's account number at the financial institution.
 - **Expires On** is only needed if the Tender Type indicates that an expiration date is necessary (e.g., for a credit card payment).
 - **Name** is the customer's name in the financial institution's system.
 - **Bill ID** is the ID of the bill whose completion caused the creation of the automatic payment. This information only appears if the automatic payment was created as a result of a bill. This information is display-only.

- If the payment was uploaded, the following fields MAY contain information (if they were populated in the interface file):
 - **MICR ID** is the value of the magnetic ink character recognition (MICR) line on the payment.
 - **Customer ID** is the customer's account ID that appeared on the interfaced payment.
 - **Ext. Reference ID** is the unique identifier of the payment upload interface record.
 - **Name** is the customer's name on the interfaced payment record.
- In the **Tender Control** area, the **Tender Control ID** is the identity of the remittance processor batch or cash drawer bundle in which the tender was remitted. Refer to [Tender Management and Workstation Cashiering](#) for more information about tender controls.

Displayed adjacent to **Tender Control** is the date, tender source, and status of the tender control.

Every new tender must reference an **open** tender control. The system will attempt to default an appropriate tender control as follows:

- If the user has at least one **open**, user-specific Tender Control, the system will default one, at random.
- If the user has no **open**, user-specific Tender Controls; the system will default an open, "all users" tender control if it can find one whose tender source indicates it's used for online cashiering. If multiple exist, one will be selected at random.

Turn off **Included in Tender Ctl Balance** if the tender should not be included in the tender amount in the tender control's tender balance. You would turn this switch off if you canceled a tender because it was mistakenly applied to the wrong account (or the amount was wrong). This switch is only enabled if:

- The tender control isn't **balanced**.
- The tender is **canceled**.

Deposit Control contains a concatenation of the tender control's deposit control's creation date, tender source type, and status. Refer to [Tender Management and Workstation Cashiering](#) for more information about deposit controls.

- The **Tender Action** area contains a button that you use to cancel a tender. Refer to [Tender Actions](#) for information about this button. Refer to [How To Cancel A Tender](#) for more information.

The **Characteristics** collection on the Tenders page contains information that describes miscellaneous information about the tender. The following fields display:

Characteristic Type	Indicate the type of characteristic.
Sequence	Controls the order in which characteristics of the same type are displayed.
Characteristic Value	Indicate the value of the characteristic.

Payment Event - Characteristics

To update payment event characteristics, open **Financial, Payment Event** and navigate to the **Characteristics** tab.

Description of Page

The **Characteristics** collection contains information that describes miscellaneous information about the payment event.

The following fields display:

Characteristic Type	Indicate the type of characteristic.
Sequence	Controls the order in which characteristics of the same type are displayed.
Characteristic Value	Indicate the value of the characteristic.

Payment Event Action Codes

The topics that follow describe each of the actions that appear on the payment event pages.

Contents

[Payment Actions](#)
[Payment Event Actions](#)
[Tender Actions](#)

Payment Actions

The topics that follow describe payment-oriented actions. Refer to [How To](#) for a description of several business processes that use these buttons.

Contents

[Distribute \(Payments\)](#)
[Redistribute \(Payments\)](#)
[Print](#)
[Freeze \(Payments\)](#)

Distribute (Payments)

Clicking the **Distribute** button distributes all **Incomplete**, **Error**, or **Freezable** payments linked to the event. Refer to [Distribute \(A Payment\)](#) for information of how a single payment is distributed.

This button is enabled if at least one payment is **Incomplete**, **Error**, or **Freezable**.

Redistribute (Payments)

Clicking the **Redistribute** button redistributes all **Incomplete**, **Error**, or **Freezable** payments linked to the event. Refer to [Redistribute \(A Payment\)](#) for information of how a single payment is redistributed.

This button is enabled if at least one payment is **Incomplete**, **Error**, or **Freezable**.

Print

This button is enabled under the following conditions:

- The document **cm_POSPrint.js** must be present on the web server and the variable defined in that document must be set to **true**. This is to indicate that point of sale (POS) printers are in use. Refer to the installation guide for more information about setting up receipt printers.

- The payment event must have at least one payment that is **frozen** AND no payments that are in **Incomplete**, **Error** or **Freezable**

Refer to [How To Print Receipts And Endorsements](#) for more information.

Freeze (Payments)

Clicking the **Freeze** button causes all **Freezable** payments linked to the event to become **frozen**. Refer to [Freeze \(A Payment\)](#) for information of how a single payment is frozen.

This button is enabled if at least one payment is **Freezable**.

If problems are detected after freezing. A payment may not be changed after it is **frozen**. All subsequent changes must occur by canceling the frozen payment and creating a new one. You can cancel and redistribute a payment on the next tab. Refer to [Canceling A Tender Versus Canceling a Payment](#) for more information.

Payment Event Actions

The topics that follow describe payment event-oriented actions. Refer to [How To](#) for a description of typical business processes that use these buttons.

Contents

- [Transfer](#)
- [Delete A Payment Event](#)

Transfer

If the payment event was created using distribution rule(s), any transfer of its payments should also be done using the distribution rules method. Hence, a different dialog opens when you click **Transfer** based on whether the payment event is associated with at least one distribution detail or not.

Contents

- [Transfer \(No Distribution Details Exist\)](#)
- [Transfer \(Distribution Details Exist\)](#)

Transfer (No Distribution Details Exist)

The **Transfer** button is enabled if:

- There is a single non-cancelled tender linked to the event AND
- There is a single frozen payment linked to the event AND
- The account on the tender is the same as the account on the payment.

When clicked, the **Payment Event Transfer** dialog opens.

You must specify the following parameters in the **Payment Event Transfer** dialog to transfer a payment:

- **Account ID** is the account to which the payment event should be transferred.
- **Cancel Reason** defines why you are performing the transfer.

- **Match Type** and **Match Value** are defaulted based on their values for the payment event that you are transferring. If necessary, modify the values appropriately for the account to which you are transferring the payment event. Match Type and Match Value are used for an account that belongs to an [open item](#) customer class or to restrict the distribution of the payment to a specific service agreement. Refer to [Payment Event - Main](#) for more information.
- Turn on **Freeze Payment** if the transferred payment should be frozen automatically. You may want to leave this turned off if you want to examine the payment distribution before freezing the payment.
- **Non CIS Name**, **Reference Number**, and **Non CIS Comments** appear if the **Account ID** belongs to a customer class that is used for non-CIS payments. Refer to [Setting Up Customer Classes](#) for more information.

Clicking **OK** causes the following to take place:

- The account on the tender is changed to reflect the transfer to account.
- The original payment and its segments are cancelled.
- A new payment is added for the transfer to account. The payment characteristics associated with the original payment are copied to the new payment.
- The new payment is distributed.
- If so designated, the new payment is frozen.

[Transfer \(Distribution Details Exist\)](#)

The **Transfer** button is enabled if:

- There is a single payor AND
- All existing payees before the transfer are the same as the payor AND
- There is at least a single non-cancelled tender linked to the event

When clicked, the **Payment Event Transfer** dialog opens.

You must specify the following parameters in this dialog to transfer payment(s):

- **Distribution Rule** is the account to which the payment event should be transferred.
- **Rule Value** is the value associated with the payment and expected by the distribution rule.
- **Amount** is the payment amount.
- **Cancel Reason** defines why you are performing the transfer.

Clicking **OK** causes the following to take place:

- The payment(s) associated with the original account are canceled.
- The new set of distribution details are processed, creating new payment(s) for the transfer to account. The payment characteristics associated with the original payment are copied to the new payment.
- If all new payments are for the same payee account (and the payee is different than the current payor) the account on the tender is changed to reflect the transfer to account.

- Original distribution details (only those with a non-zero amount) are deleted and replaced by the new set of details on the payment event.

Distribution rule entries can have a zero amount. Refer to [Rule Value Can Capture Additional Information](#) for more information about how to use distribution details to capture additional payment related information.

Determine Tender Account. This dialog does not call the *Determine Tender Account* algorithm defined on [distribution rule](#) as this action does not rebuild the payment tender collection.

Refer to [Rule Value Can Capture Additional Information](#) for more information about how to use distribution details to capture additional payment related information.

Delete A Payment Event

The **Delete** button deletes a payment event and its tenders and payments.

This button is enabled if there are no *frozen* or *canceled* payments and no *canceled* tenders.

Tender Actions

The topics that follow describe tender-oriented actions. Refer to [How To](#) for a description of several business processes that use these buttons.

Cancel A Tender

Clicking **Cancel** causes the following to take place:

- All payments associated with the tender's payment event are *canceled*.
- The tender becomes *canceled*.
- If the cancellation reason indicates the cancellation is due to non-sufficient funds, other actions may occur. Refer to [NSF Cancellations](#) for more information.

The **Cancel** button is enabled if the tender is not *canceled*

When clicked, the **Tender Cancel** window opens.

You must specify the following parameters to cancel a tender:

- Select a **Cancel Reason** to describe why the tender is being canceled.
- The system also needs to know which **Bank Account To Charge** the cancellation against. The system will default the **Bank Code** and **Bank Account** from the original bank information used when the tender was deposited. You can override them here.

When you click the **OK** button, the system cancels the tender and ALL payments linked to the event.

When non-canceled tenders are still linked to the event. If there are multiple tenders linked to the payment event, you must either add new payment(s) that equal the sum of the event's non-canceled tenders or cancel the remaining tenders.

Cancellation after cash back. If a customer tenders a check for \$100, but only owes \$80, the system will recommend returning \$20 in cash to the customer (assuming the tender type for checks allows overpayment). If the tender type for checks is not "like cash", a second tender is created for -\$20 (the first tender is for \$100). If the check subsequently bounces, both tenders must be cancelled.

Payment Event Quick Add

The Payment Event Quick Add page is used to quickly add, distribute and freeze one or more payment events using [distribution rules](#). Open this page using **Financial, Payment Event Quick Add**.

Description of Page

Specify the **Tender Control ID** in which the tenders will be recorded. Every new tender must reference an **open** tender control. The system will attempt to default an appropriate tender control as follows:

- If the user has at least one **open**, user-specific Tender Control whose type is online cashiering or ad hoc, the system will default one, at random.
- If the user has no **open**, user-specific Tender Controls; the system will default an open, "all users" tender control if it can find one whose tender source has a tender source type of online cashiering or ad hoc. If multiple exist, one will be selected at random.

Specify the **Payment Date**. The current date defaults.

Use **Number of Payment Events** to indicate whether you intend to add a **single** payment event or **multiple** payment events.

Choose the **multiple** payment events option if you intend to add "simple" payment events. By "simple" we mean:

- A single account is both the payor account and the account whose debt is being relieved by the payment.
- The payment tender is not an auto-pay tender.

For all other cases choose the **single** payment event dialogue as it is designed to handle more complex payment event configurations such as when:

- The payor account is different than the payee account.
- Multiple payors cover payments for one or more payees.
- The payment tender is an auto-pay tender.

Once you have made your selection the page is adjusted to support the corresponding dialog.

Cash-only customers. When you attempt to add a new payment event, the system warns you if the account remitting the tender is a cash-only customer. An account is considered cash-only if their current cash-only points exceed the cash-only tolerance maintained on the Installation record. A customer's cash-only points are maintained on [Account - Credit Rating](#).

Contents

[Multiple Payment Events Dialog](#)
[Single Payment Event Dialog](#)

Multiple Payment Events Dialog

This section describes the default dialog setup when navigating from **Financial, Payment Event Quick Add**.

Description of Page

Using the **multiple** payment events option, each row in the payment detail grid typically represents a unique distribution detail. However, more than one row in the grid can belong to a single payment event. All distribution details for identical tender account will be grouped under a single pay event.

The following columns appear in the **Payments** grid:

- Select a **Distribution Rule** by which the payment detail is to be processed. If you have set up a [default distribution rule](#), it is defaulted in the first row.
- Specify the **Rule Value** associated with the payment and expected by the distribution rule.
- Use **Payment Amount** to define the amount of the payment.
- Use **Tender Type** to define the form or remittance (e.g., cash, check, etc.). Note that the **Tender Type** defaults from the [installation record](#).

Authorizing Tenders. The **Tender Type** dropdown list does not include tender types that require authorization (i.e. credit card payments).

For information about tender types requiring authorization, refer to [Credit Card Payments](#).

Automatic Payments. The multiple payment events dialog does not support automatic payments. Switch to the single payment event dialog to enter automatic payments.

- Use **Check Number** if a check is remitted.
- **MICR ID** is the value of the magnetic ink character recognition (MICR) line on the payment.
- You may use **Customer ID** to record additional customer information.
- You may use **External Reference ID** to record external information associated with the payment tender.
- You may use **Name** to record additional payment tender information.

- **MICR ID** is the MICR ID associated with the tender.
- **Payor Account ID** is the tender account as determined by the [Determine Tender Account](#) algorithm defined on the [distribution rule](#). This information is only populated after the distribution detail has been processed.

After specifying the various payment distribution details in the grid, click **Create**. The system attempts to create payment event(s) as follows:

- As mentioned earlier, all distribution details for identical tender accounts are grouped under a single pay event. The first step in identifying common tenders is to determine the tender account. To do that the system calls the [Determine Tender Account](#) algorithm defined on the [distribution rule](#).
- A payment event is created for each unique tender account.
- All rows having all attributes of the tender identical except for the amount are grouped together where each group represents a single tender.
- A single payment tender is created for each unique tender and linked to the payment event.
- The total payment amount for each distinct group of distribution details (for the payment event) having the same rule and value is summarized. For each distinct group, the system calls the [Create Payment](#) algorithm defined on the [distribution rule](#) providing it with the rule value and total payment amount. It is the responsibility of this algorithm to create the payments for this payment event.
- Add the distinct payment event distribution detail and its total amount beneath the payment event.
- If the tender control's tender source type indicates that this is a cashiering station and your implementation is configured for printing at a cashier station, the [Print Dialog](#) appears.

The **Payment Event ID** column appears in the grid after the distribution details have been processed. It shows the payment event ID created for the distribution detail record and a short description as to the status of the payment(s) created for the payment event. You can use this field to navigate to the payment event.

After you've added a group of payment events, you should press the [clear](#) button (or Alt-C), to ready the page for the next group of payment event details.

Single Payment Event Dialog

If you have opted to always use the payment event distribution rules method as your [default method](#), this section describes the default dialog setup when you navigate to the **Payment Event** page in **Add** mode.

Adding a **single** payment event at a time, allows for more complex payment event information to be captured.

Enter one row in the **Tenders** grid for every unique tender associated with the payment event.

- **Payor Account ID** references the tendering account. The name of the account's main customer is displayed adjacent. If the account has exceeded your cash-only threshold, the warning **Cash only** appears.

Warning! If you change the Payor Account ID and the tender has an associated automatic payment request, the automatic payment request will be removed. A new automatic payment request will be created for the new Payor Account ID. Refer to [Automatic Payments](#) for more information.

- **Tender Amount** is the amount of the tender.
- Use **Tender Type** to define the form or remittance (e.g., cash, check, etc.). Note, the **Tender Type** defaults from the [installation record](#).
- Use **Check Number** if a check is remitted.
- **MICR ID** is the value of the magnetic ink character recognition (MICR) line on the payment.
- You may use **Customer ID** to record additional customer information.
- You may use **External Reference ID** to record external information associated with the payment tender.
- You may use **Name** to record additional payment tender information.
- If the **Tender Type** is associated with an automatic payment, the page displays a section that includes information about how and when the automatic payment was interfaced to the payment source.
- If the **Tender Type** is associated with an [automatic payment](#), the system attempts to default automatic payment information from the [account's auto-pay](#) option if the tender type is the same as the tender type on the account's auto-pay source and if the auto pay option is effective on the payment date. If the system is unable to default information, you must specify the source of the funds and the customer's account number / credit card number at the financial institution.
 - **Auto Pay Source Code** is the financial institution / credit card company that receives the automatic payment request.
 - **External Account ID** is the customer's account number at the financial institution.
 - **Expires On** is only needed if the **Tender Type** indicates that an expiration date is necessary (e.g., for a credit card payment).
 - **Name** is the customer's name in the financial institution's system.

The following columns appear in the **Payments** grid:

- Select a **Distribution Rule** by which the payment detail is to be processed. If you have set up a [default distribution rule](#), it is defaulted in the first row.
- Specify the **Rule Value** associated with the payment and expected by the distribution rule.
- Use **Payment Amount** to define the amount of the payment.

After specifying the various payment distribution details in the grid, click **Create**.

Go To Payment Event. If you wish to be transferred to the payment event page once processing is complete, remember to check the Go To Payment Event check box before you click **Create**. This field does not appear when this page is used as the default Payment Event - Add Dialog and you are automatically transferred to the payment event page once the payment event is created.

After distribution details have been processed, a short description of the payment event that has been created is displayed to the right of the **Go To Payment Event** field. The description provides information as to the status of the payment(s) created for the payment event. You can use this field to navigate to the payment event.

The system attempts to create the payment event as follows:

- Validate that the sum of the tenders is not less than the sum of the payments.
- Create the payment event with a separate payment tender for each row in the Tenders grid.
- Summarize the total payment amount for each distinct group of distribution details having the same rule and value. For each distinct group, call the **Create Payment** algorithm defined on the [distribution rule](#) providing it with the rule value and total payment amount. It is the responsibility of this algorithm to create the payments for the payment event.
- Add the distinct payment event distribution detail summary beneath the payment event.
- If the tender control's tender source type indicates that this is a cashiering station and your implementation is configured for printing at a cashier station, the [Print Dialog](#) appears.

Determine Tender Account. Having an explicit collection of tenders eliminate the need to determine the tender account. Therefore this dialog does not call the **Determine Tender Account** algorithm.

Payment Quick Add

The Payment Quick Add page is used to quickly add, distribute and freeze payment events for up to fifteen accounts. Open this page using **Financial, Payment Quick Add**.

Description of Page

Specify the **Tender Control ID** in which the tenders will be recorded. Every new tender must reference an **open** tender control. The system will attempt to default an appropriate tender control as follows:

- If the user has at least one **open**, user-specific Tender Control whose type is online cashiering or ad hoc, the system will default one, at random.
- If the user has no **open**, user-specific Tender Controls; the system will default an open, "all users" tender control if it can find one whose tender source has a tender source type of online cashiering or ad hoc. If multiple exist, one will be selected at random.

Specify the **Payment Date**. The current date defaults.

Enter one row in the grid for every payment event to be added. The following information is entered for each payment event:

- Use **Account ID** to define the customer who tendered the payment. It's important to note that this transaction assumes that the tendering account is the same as the account whose balance is relieved by the payment. Refer to [How To Allocate The Tender Amount To Multiple Accounts](#) if you need to distribute the payment to an account (or accounts) other than the tendering account.

Cash-only customers. You cannot use this transaction to add a payment for a "cash-only" account if the tender type is not "like cash". An account is considered cash-only if its cash-only points exceed the cash-only tolerance maintained on the Installation record. A customer's cash-only points are maintained on [Account - Credit Rating](#). You must use the [Payment Event transaction](#) to add such a payment (note, if you use the Account context menu to transfer to the Payment Event page in add mode, you won't have to retype the Account ID when you add the payment event).

- Use **Payment Amount** to define the amount of the payment.
- Use **Tender Type** to define the form or remittance (e.g., cash, check, etc.). Note, the **Tender Type** defaults from the [installation record](#).

Authorizing Tenders. The **Tender Type** dropdown list does not include tender types that require authorization (i.e. credit card payments).

For information about tender types requiring authorization, refer to [Credit Card Payments](#).

- Use **Check Number** if a check is remitted.
- **Match Type** and **Match Value** are only used if either of the following conditions is true:
 - This **Account ID** belongs to an [open item](#) customer class. In this situation, specify a **Match Type** to define how the payment should be matched to the customer's open-items and use **Match Value** to define the open-items covered by the payment. For example, if this payment is in respect of a bill, specify a match type of "bill id" and a match value of the bill id being paid.

Shortcut. If you enter a **Match Type** of "bill id" and leave the **Match Value** blank, we assume the customer wants to pay the latest bill.

- The customer wants to restrict the distribution of the payment to a specific service agreement. In this situation, specify a **Match Type** of "service agreement ID" and a **Match Value** of the respective service agreement ID.

After specifying the various accounts and amounts in the grid, click the **Distribute and Freeze** button. When you click this button, the system attempts to create a payment event for each row in the grid. Four potential outcomes are possible for each row:

- If the data you entered for a payment event isn't complete (e.g., you don't specify a valid account or amount):
 - No payment event is created.
 - A **Message** describing the problem is displayed.
 - All fields on the row remain modifiable.

You should correct each such line and then press the **Distribute and Freeze** button.

- If the data you entered is complete, but the system issues a warning (e.g., the account is considered "cash-only" and a tender type other than cash is remitted, or if the account already has a payment for this amount on this date):

- No payment event is created.
- A **Message** containing the warning is displayed.
- All fields on the row remain modifiable.

You must clear the line and add a payment using the [Payment Event transaction](#) (note, if you use the Account context menu to transfer to the Payment Event page in add mode, you won't have to retype the Account ID when you add the payment event).

- If the data you entered is complete, but the system is not successful in distributing the payment:
 - The system creates a payment event, a payment and a tender.
 - A **Message** describing the problem is displayed.
 - The payment's **Status** is displayed
 - All fields on the row are disabled.

You must press the adjacent go to button to drill to the payment event where the error can be corrected.

- If the data you entered is complete and the system is successful in distributing the payment:
 - The system creates a payment event, a payment, and a tender.
 - The system distributes the payment amongst the account's service agreements and then freezes the payment.
 - The payment's **Status** is displayed.
 - All fields on the row are disabled.
 - If the tender control's tender source type indicates that this is a cashiering station and your implementation is configured for printing at a cashier station, the [Print Dialog](#) appears.
- You can press the adjacent go to button to view a payment event.

Separate Commits. This page is unusual in that each payment event is committed to the database independently. For example, if you enter seven payment events and one is invalid, six payment events will be added to the database when you press **Distribute and Freeze**. When the page is redisplayed, the rows containing the valid payments are protected and an indication of their validity is displayed. The row containing the invalid payment remains unprotected. You can correct the erroneous payment and then press the **Distribute and Freeze** button again.

After you've added a group of payments, you should press the [clear button](#) (or Alt-C), to ready the page for the next group of payment events.

Maintaining Payments

Payments and payment segments are automatically created by the system when [payment events](#) are created. You should only need to access this transaction as follows:

- To view a payment's payment segments (i.e., to view how a payment was distributed amongst its account's service agreements).

- To override the distribution of a payment amongst its account's service agreements. Refer to [Distributing A Payment Amongst An Account's Service Agreements](#) for information describing how payments are typically distributed.

The topics in this section describe how to maintain a payment.

Contents

[Payment - Main](#)
[Payment - Pay Segments](#)
[Payment - Manual Distribution](#)
[Payment - Characteristics](#)
[Payment Action Codes](#)

Payment - Main

This page contains basic information about a payment. Open this page using **Financial, Payment, Main**.

Most payments are maintained using the [Payment Event](#) page. The transaction described below is typically only used to view a payment's payment segments and to manually distribute a payment amongst an account's service agreements.

The **Description of Page** section below describes the fields on this page. Refer to [How To](#) for instructions on how to perform common maintenance functions.

Description of Page

Payment Info contains a concatenation of the payment date, amount, status, and the name of the main customer on the account.

Payment ID is the system-assigned unique identifier of the payment.

Payment Event ID is the system-assigned unique identifier of the payment event. The adjacent info contains a concatenation of the payment date, amount, and the name of the main customer on the account that remits the event's tender. If multiple tenders exist, the customer's name is not displayed. A warning may also be displayed. The possible warnings are **Unfrozen Payments** and **The Payment Event is unbalanced**.

Payment(s) contains the total number and value of payments linked to the payment event.

Tender(s) contains the total number and value of tenders linked to the payment event.

Account ID is the account whose debt is reduced by the payment. The name of the account's main customer is displayed adjacent. This field is gray after the payment is frozen.

Payment Amount is the amount of the payment. This field is gray after the payment is frozen. If the account has exceeded your [cash-only threshold](#), a warning (in red) appears adjacent to **Payment Amount**. Note, a customer's cash-only points are maintained on [Account - Credit Rating](#).

Payment Status is the payment's status. Refer to [Payment Lifecycle](#) for the potential values and how to handle a payment when it exists in a given state. If the payment is in **Error**, the error message is displayed adjacent. If the payment is **canceled**, the **Cancel Reason** is displayed adjacent.

Match Type and **Match Value** will only be used if either of the following conditions is true:

- This **Account ID** belongs to an [open item](#) customer class. In this situation, specify a **Match Type** to define how the payment should be matched to the customer's open-items and use **Match Value** to define the open-items covered by the payment. For example, if this payment is in respect of a bill, specify a match type of "bill id" and a match value of the bill id being paid.
- The customer wants to restrict the distribution of the payment to a specific service agreement. In this situation, specify a **Match Type** of "service agreement ID" and a **Match Value** of the respective service agreement ID.

Match Type and **Match Value** are gray after the payment is frozen.

If the payment is linked to an **Account ID** that belongs to a [customer class](#) that is used for non-CIS payments, the following fields appear (note, these fields are gray after the payment is frozen):

- **Name** is the name of the person remitting the payment.
- **Reference Number** is the reference number of the item being paid (e.g., the property tax reference number).
- **Comments** are used to describe anything unusual about the non-CIS payment.

The contents of the **Payment Segments** section depends on the number of payment segments linked to the payment. If more than 25 payment segments exist, a message appears that allows you to navigate to [Payment - Payment Segments](#) where you can view the payment segments (this tab page has special functionality that allows you to defined which payment segments should be displayed). If 25 or fewer payment segments exist, the grid contains information about each payment segment:

- **Premise** is the address of the service agreement's main premise.
- **SA Information** contains basic information about the service agreement.
- **Distributed Amt** is the amount of the service agreement's debt relieved by the payment.

The following information appears at the bottom of the page:

- **Payment Amount** is the amount of the payment.
- **Distributed Amount** is the sum of the payment segments linked to the payment.
- The **Difference** between the **Payment Amount** and the **Distributed Amount** appears if it is non-zero.
- **Billed Amount** is the amount of the bill preceding the payment date.
- **Delinquent Amount** is the amount of the customer's debt that was due on / before the prior bill's due date.
- **Current Balance** is the account's current balance.

Refer to [Payment Action Codes](#) for information about the action buttons on this page.

Payment - Pay Segments

You can use this page to view all or selected payment segments linked to a payment.

Open **Financial**, **Payment**, **Payment Segments** to view this information.

Most payments are maintained using the [Payment Event](#) page. The transaction described below is typically only used to view a payment's payment segments.

Note. If the payment has more than 25 payment segments, the search criteria are intentionally left blank in order to avoid retrieving all payment segments (with the resultant slow response times). You must therefore use the **SA Filter** to define the type of payment segments that should be retrieved. See the [Description of page](#) below for more information about this page's search criteria.

Description of page

Payment Info contains a concatenation of important information about the payment.

Payment ID is the system-assigned unique identifier of the payment.

Payment Event ID is the system-assigned unique identifier of the payment event. The adjacent info contains a concatenation of the payment date, amount, and the name of the main customer on the account that remits the event's tender. If multiple tenders exist, the customer's name is not displayed. A warning may also be displayed. The possible warnings are [Unfrozen Payments](#) and [The Payment Event is unbalanced](#).

Payment(s) contains the total number and value of payments linked to the payment event.

Tender(s) contains the total number and value of tenders linked to the payment event.

Note. If the payment has more than 25 payment segments, the search criteria are intentionally left blank in order to avoid retrieving all payment segments (with the resultant slow response times). You must therefore use the **SA Filter** to define the type of payment segments that should be retrieved.

Use the **SA Filter** to define the types of service agreements whose payment segments appear in the grid. The following options are available:

- **Address.** Use this option to restrict payment segments to those whose service agreements are 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 payment segments based on service agreement attributes.
- **Geographic Type.** Use this option to restrict payment segments to those whose service agreements are linked to service points associated with a given **Geographic Type** and **Geographic Value**.
- **SA Type.** Use this option to restrict payment segments to those whose service agreements are linked to a given **CIS Division** and **SA Type**.

Don't forget to click the search button after changing the **SA Filter**.

The grid that follows contains the payment segments that match your search criteria. The following information appears in the grid:

- The **Premise** column contains the characteristic premise associated with the payment segment's service agreement. Refer to [Maintaining Premises](#) for more information about this premise.

- Click the go to button to transfer to view the financial transaction related to the payment segment. This is only available after the payment has been frozen.
- **SA Information** contains summary information about the service agreement.
- **Distributed Amt** is the amount of the service agreement's debt relieved by the distribution.
- **Adjustment ID** is displayed if there is an adjustment associated with the payment. Refer to [Loan Overpayments](#) for an example of when an adjustment is associated with a payment.

The following information appears at the bottom of the page:

- **Payment Amount** is the amount of the payment.
- **Distributed Amount** is the sum of the payment segments linked to the payment.
- The **Difference** between the **Payment Amount** and the **Distributed Amount** appears if it is non-zero.
- **Billed Amount** is the amount of the bill preceding the payment date.
- **Delinquent Amount** is the amount of the customer's debt that was due on / before the prior bill's due date.
- **Current Balance** is the account's current balance.

Payment - Manual Distribution

You can use this page to view service agreements linked to the payment's account and to manually distribute a payment amongst specific service agreements.

Open this page using **Financial, Payment, Manual Distribution**.

Most payments are maintained using the [Payment Event](#) page. The transaction described below is typically only used to view a payment's payment segments and to manually distribute a payment amongst an account's service agreements.

Note. If the payment's account has more than 25 service agreements, the search criteria are intentionally left blank in order to avoid retrieving all service agreements (with the resultant slow response times). You must therefore use the **SA Filter** to define the type of service agreements that should be retrieved. See the [Description of page](#) below for more information about this page's search criteria.

Description of page

Payment Info contains a concatenation of important information about the payment.

Payment ID is the system-assigned unique identifier of the payment.

Note. If the payment's account has more than 25 service agreements, the search criteria are intentionally left blank in order to avoid retrieving all service agreements (with the resultant slow response times). You must therefore use the **SA Filter** to define the type of service agreements that should be retrieved.

Use the **SA Filter** to define the types of service agreements that appear in the grid. The following options are available:

- **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 want to view all service agreements.
- **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**.

Don't forget to click the search button after changing the **SA Filter**.

The grid that follows contains the service agreements that match your search criteria. The following information appears in the grid:

- **Premise** contains the characteristic premise associated with the service agreement.
- **SA Information** contains summary information about the service agreement.
- **Distributed Amt** is the amount of the service agreement's debt relieved by the payment. This field is gray after the payment is frozen.
- **Billed Amt** is the amount that was billed for the service agreement on the bill preceding the payment date.
- **Delinquent Amt** is the amount of debt associated with financial transactions that appear on overdue bills.
- **Current Balance** is the amount currently owing for this service agreement.

If the account being paid is an **open item** account, then an alternate grid is displayed. The alternate grid is designed to break down payable balances by individual bill and service agreement. Both unpaid bill balances and new charges are displayed. After distribution, a separate match event will be created for each bill paid or partially paid.

Note that an implementation can override the type of grid displayed for open item accounts by setting up a [Financial Transaction Options Feature Configuration](#) with an **Always Use Manual Payment Distribution By SA** option set to **Y**. With this in place, the system will always display the manual distribution by service agreement grid.

The open item manual distribution grid contains the following columns:

- **Bill Date**
 - Shows bill date if the row contains an account's unpaid bill.
 - Shows '**New Charges**' if the row contains an account's unbilled new charges.
 - Shows '**Payment Segment**' for the following conditions:
 - Row is a pay segment of a cancelled payment.

- Row is a pay segment of an overpayment to the highest priority SA or an excess credit SA.
 - Row is a pay segment of a balance forward payment distribution.
- **Premise Information** contains the characteristic premise associated with the service agreement.
- **SA Information** contains summary information about the service agreement.
- **Distributed Amount**
 - You can enter an amount to distribute if the payment is non-frozen and non-cancelled, otherwise, this field is protected.
- **Billed Amount**
 - The amount that was billed for the service agreement on the row's bill. If the row does not represent a bill amount then this will be zero.
- **Unpaid Amount**
 - This is determined by the "determine open-item bill amount" algorithm as defined on the installation option.
 - This column will contain zero if:
 - The row is linked to an open non-disputed match event with other bills' FTs on it. In this case, it is not possible to determine the bill amount.
 - The row is linked to an open disputed match event.

For either grid, the following information appears at the bottom of the page:

- **Payment Amount** is the amount of the payment.
- **Distributed Amount** is the sum of the payment segments linked to the payment.
- The **Difference** between the **Payment Amount** and the **Distributed Amount** appears if it is non-zero.
- **Billed Amount** is the amount of the bill preceding the payment date.
- **Delinquent Amount** is the amount of the customer's debt that was due on / before the prior bill's due date.
- **Current Balance** is the account's current balance.

Payment - Characteristics

To update payment characteristics, open **Financial, Payment** and navigate to the **Characteristics** tab.

Description of Page

The **Characteristics** collection contains information that describes miscellaneous information about the payment. The following fields display:

Characteristic Type	Indicate the type of characteristic.
Sequence	Controls the order in which characteristics of the same type are displayed.

Characteristic Value

Indicate the value of the characteristic.

Payment Action Codes

The topics that follow describe each of the actions that appear on the payment pages.

Contents

- [Distribute \(A Payment\)](#)
- [Redistribute \(A Payment\)](#)
- [Freeze \(A Payment\)](#)
- [Cancel \(A Payment\)](#)
- [Transfer \(A Payment\)](#)
- [Delete \(A Payment\)](#)

Distribute (A Payment)

Clicking the **Distribute** button causes a payment to be distributed amongst the account's service agreements. Refer to [Distributing A Payment Amongst An Account's Service Agreements](#) for a description of how this is achieved.

This button is enabled if a payment is **Incomplete**, **Error**, or **Freezable**. Refer to [Payment Lifecycle](#) for more information about these status values.

If the payment is distributed appropriately, click the [Freeze](#) button to freeze the payment and its payment segments.

Redistribute (A Payment)

Clicking the **Redistribute** button causes a payment to be redistributed amongst the account's service agreements. Refer to [Distributing A Payment Amongst An Account's Service Agreements](#) for a description of how this is achieved.

This button is enabled if a payment is **Incomplete**, **Error**, or **Freezable**. Refer to [Payment Lifecycle](#) for more information about these status values.

If the payment is distributed appropriately, click the [Freeze](#) button to freeze the payment and its payment segments.

Note. This button is only available on the [Payment - Main](#) and [Payment - Manual Distribution](#) pages.

Freeze (A Payment)

Clicking the **Freeze** button causes a payment and its payment segments to become **frozen**. Refer to [Payment Lifecycle](#) for more information about freezing.

This button is enabled if the payment is **Freezable**.

If problems are detected after freezing. A payment may not be changed after it is **frozen**. All subsequent changes must occur by canceling the frozen payment and creating a new one.

Cancel (A Payment)

Clicking the **Cancel** button causes the payment and its payment segments to be **canceled**. Canceling a payment (as opposed to canceling a tender) is typically only done if the original distribution was frozen and it's incorrect.

This button is enabled if the payment is **Frozen**. Refer to [Payment Lifecycle](#) for more information about these status values.

Transfer (A Payment)

The **Transfer** button is enabled if the payment is **Frozen**. Refer to [Payment Lifecycle](#) for more information about these status values.

If the payment being transferred was created using payment event distribution rule(s), the payment transfer should also be done using the distribution rules method. Hence, a different dialog opens when you click **Transfer** based on whether the payment's payment event is associated with at least one distribution detail or not.

Contents

[Transfer Payment \(No Distribution Details Exist\)](#)

[Transfer Payment \(Distribution Details Exist\)](#)

Transfer Payment (No Distribution Details Exist)

Payment transfer causes the payment and its payment segments to be **canceled** and a new payment to be created. Transferring a payment is typically only done if the original distribution was targeted at the wrong account but all tender information for the pay event is correct. Unlike the [payment event transfer](#), transferring a single payment does not affect the pay event's tender information.

You must specify the following parameters in the **Payment Transfer** window to transfer a payment:

- **Account ID** is the account to which the payment should be transferred.
- **Cancel Reason** defines why you are performing the transfer.
- **Match Type** and **Match Value** are defaulted based on their values for the payment event that you are transferring. If necessary, modify the values appropriately for the account to which you are transferring the payment event. Match Type and Match Value are used for an account that belongs to an [open item](#) customer class or to restrict the distribution of the payment to a specific service agreement. Refer to [Payment Event - Main](#) for more information.
- Turn on **Freeze Payment** if the transferred payment should be frozen automatically. You may want to leave this turned off if you want to examine the payment distribution before freezing the payment.
- **Non CIS Name**, **Reference Number**, and **Non CIS Comments** appear if the **Account ID** belongs to a customer class that is used for non-CIS payments. Refer to [Setting Up Customer Classes](#) for more information.

Clicking **OK** causes the following to take place:

- The original payment and its segments are cancelled.
- A new payment is added for the transfer to account. The payment characteristics associated with the original payment are copied to the new payment.

- The payment is distributed amongst the new account's service agreements.
- If so designated, the new payment is frozen.

Transfer Payment (Distribution Details Exist)

Payment transfer causes the payment to be **canceled** and new payment(s) to be created using new payment event distribution rule(s). Transferring a payment is typically done when a payment event has multiple payments and one or more payments have been assigned to the wrong SA. Each of the incorrectly directed payments can be transferred to the correct SA(s) using the **Transfer via Distribution Rules** dialog.

Total Amount is the total payment amount to transfer from.

You must specify the following parameters in order to transfer a payment:

- **Distribution Rule** is the service agreement to which the payment should be transferred.
- **Rule Value** is the value associated with the payment and expected by the distribution rule.
- **Amount** is the payment amount.
- **Cancel Reason** defines why you are performing the transfer.

Clicking **OK** causes the following to take place:

- The payment associated with the original account is canceled.
- The new set of distribution details is processed, creating new payment(s) for the transfer-to service agreement(s). The payment characteristics associated with the original payment are copied to the new payment.
- The new set of distribution details is added to the payment event. The original distribution details are kept.

Distribution rule entries can have a zero amount. Refer to [Rule Value Can Capture Additional Information](#) for more information about how to use distribution details to capture additional payment related information.

Determine Tender Account. This dialog does not call the **Determine Tender Account** algorithm defined on [distribution rule](#) as this action does not rebuild the payment tender collection.

Delete (A Payment)

The **Delete** button deletes a payment. This button is enabled if the payment is **Incomplete**, **Error**, or **Freezable**.

Payment Portal

The base package is shipped with a configured payment portal that can be used for payment processing. The portal supports the following functionality that is not supported in the base [Payment Event - Add Dialog](#):

- Processing of payments that require authorization (i.e. credit card payments)
- Creating non CIS payments based on payment templates
- Processing payment tenders in alternate currencies

Refer to [The Big Picture of Portals and Zones](#) for a description of portal and zone functionality.

Contents

[Credit Card Payments](#)

[Non CIS Payments](#)

[Alternate Currency Payments](#)

Credit Card Payments

The payment portal provides the functionality needed for posting and authorizing customers' credit card charges in real-time. If the credit card authorization is successful, the payment will be added automatically (assuming you selected the appropriate payment freeze action via the portal).

Refer to [How To](#) for instructions describing how to add a payment that requires authorization.

Non CIS Payments

The payment portal allows you to add non-CIS payments using a payment template. Payment Templates can be configured for common types of payment allocations to pre-populate the payment distribution, allowing non-CIS payments to be directly allocated against the appropriate distribution codes.

Refer to [How To](#) for instructions describing how to add a non-CIS payment using a payment template.

Alternate Currency Payments

The payment portal allows you to process a payment tender remitted in an alternate currency. The system enables conversion of the tendered amount into the account's currency.

Refer to [How To](#) for instructions describing how to process a payment in an alternate currency.

How To

The topics in this section describe how to perform common payment maintenance functions.

Contents

[How To Add A New Payment Event](#)

[How To Cash A Check](#)
[How To Allocate The Tender Amount To Multiple Accounts](#)
[How To Print Receipts And Endorsements](#)
[How To Cancel A Tender](#)
[How To Transfer A Payment From One Account To Another](#)
[How To Distribute A Payment To A Specific Service Agreement](#)
[How To Get An Unbalanced Tender Control In Balance \(Fixing Over/Under\)](#)
[How To Add A Non-CIS Payment Using A Template](#)
[How To Authorize A Credit Card Tender](#)
[How To Reverse A Credit Card Tender](#)
[How To Process An Alternate Currency Payment](#)

How To Add A New Payment Event

There are several ways to add a new event:

- You can use [Payment Quick Add](#) to quickly add one or more payment events. You'd use this approach to add simple payments where no manual intervention is required. By "simple payment" we mean:
 - The account is both the tendering account and the account whose debt is being relieved by the payment
 - The payment date is the current date
 - The payment should be [distributed](#) amongst the account's service agreements using standard distribution logic
- If applicable to your business practice, you can use [Payment Event Quick Add](#) to quickly add one or more payment events using [distribution rules](#).
- You can use [Payment Event Maintenance](#) to add a payment event. You would use this approach if multiple forms of payment are remitted (e.g., cash and a check) or if there are multiple payors and/or payees linked to the payment event.

Note. By default the system opens the [Payment Event - Add Dialog](#) when you navigate to this page in add mode. If you have opted to always use the payment event distribution rules method as your [default method](#), the [Payment Event Quick Add \(Single Payment Event\)](#) page appears instead. Refer to these pages for more information.

How To Cash A Check

If you allow customers to cash checks, you'll have a payment event with two tenders:

- The first tender contains information about the check.
- The second tender contains information about the cash refund (the tender amount is a negative amount equal to the amount of the check).

The interesting aspect of this payment event is that it has no payments because the sum of the tenders is zero.

We start this explanation in the middle of [How To Add A New Payment Event](#).

In the **Payment Event Add** dialog, be sure to select **Do Not Distribute** before clicking **OK**.

- On the **Main** page, click – to remove the payment (cashing a check is a payment event with no payments).
- Transfer to the **Payment Event, Tenders** page.
- **Insert** a row in the tender scroll (click the + button) for the cash refund tender. In this row enter a tender type of “cash” and a tender amount of negative the check amount.
- Save the event (note there is nothing to distribute or freeze).

How To Allocate The Tender Amount To Multiple Accounts

When you add a payment event, the system automatically creates a single payment for the account that remitted the funds. If someone is remitting funds for someone other than themselves, you must change/add payments. This section describes how to do this.

Note. This section assumes you chose to add the payment event using the [Payment Event - Add Dialog](#). Refer to [How To Add A New Payment Event](#) for the complete list of options.

In the [Payment Event - Add Dialog](#), be sure to select **Do Not Distribute** before clicking **OK**.

Once on the payment event page, go to the Tenders tab and do the following:

- If the remitting account shouldn't have received any part of the payment, **Remove** it by clicking the - button. Alternatively, you can just change the account id to reflect the recipient.
- If multiple accounts receive the remitted funds, **Insert** one row in the payment scroll (click the + button) for each additional account.
- When the payment event is balanced, **Save** it. Then **Distribute** and **Freeze** the payment(s). Refer to [Payment Event Actions](#) for more information on these action buttons.

How To Print Receipts And Endorsements

The system can be configured to allow you to endorse checks and print receipts using special cashiering station printers. This section describes the print options available.

Note. These options are only available if they have been installed in your system. Installing these options are a delivery and installation issue and are not within the domain of this document.

The print functions are available from the [Payment Event](#), [Payment Quick Add](#) and [Payment Event Quick Add](#) transactions. Refer to those sections for more information.

The print dialog functions are described below.

Description of Page

Use the **Endorse** button to endorse checks using the cashier station printer. This is only enabled if one of the tender types indicates a check. If there were multiple checks, use the scroll buttons to select them. Feed the appropriate check in the printer and an endorsement message is printed on the back.

Use the **Receipt** button to print a receipt using the cashier station printer. Choose a **Short** or **Long** option. Use the **Duplicate** button to print a duplicate receipt – this prints the receipt with a “duplicate” label.

Use the **Stub** button to print special information in bill stub format (account, name, amount). Feed a slip of paper into the printer and the special stub information will be printed.

When you are finished, click **Done**; click **Cancel** to exit at any time.

The various text strings that are printed on the receipt and the endorsement are defined on [Installation Options - Messages](#).

How To Cancel A Tender

If a tender is no longer valid (e.g., a check bounces), the tender must be canceled. The following steps explain how to do this.

Warning! These steps only make sense in the context of the page used to maintain tenders. Refer to [Payment Event - Tenders](#) for the details.

You navigate through the pages as follows:

1. Open the payment event - tenders page for the tender in question. Proceed to step 2.
2. Click **Cancel**. Proceed to step 3.
3. Before the system cancels the tender, you must specify the cancel reason. The tender cancel window opens.

In the tender cancel window, select a **Cancel Reason** to describe why the tender is being canceled. The system also needs to know which **Bank Account To Charge** the cancellation against. The system will default the **Bank Code** and **Bank Account** from the original bank information used when the tender was deposited. You can override them here.

When you click the **OK** button, the system cancels the tender and ALL payments linked to the event.

If the cancellation reason you supply is indicated as being “non-sufficient funds”, the system will generate an adjustment to levy a NSF charge. Refer to [SA Type - Main Information](#) for more information.

When non-canceled tenders are still linked to the event. If there are multiple tenders linked to the payment event, you must either add new payment(s) that equal the sum of the event's non-canceled tenders or cancel the remaining tenders.

Cancellation after cash back. If a customer tenders a check for \$100, but only owes \$80, the system will recommend returning \$20 in cash to the customer (assuming the tender type for checks allows overpayment). If the tender type for checks is not “like cash”, a second tender is created for -\$20 (the first tender is for \$100). If the check subsequently bounces, both tenders must be cancelled.

How To Transfer A Payment From One Account To Another

If you need to transfer a payment amount from one account to another, you can use the **Transfer** button either on the payment event page or the payment page.

Refer to [Transferring A Payment](#) for a description of when to use each option.

How To Distribute A Payment To A Specific Service Agreement

When you click **Distribute**, the system uses the payment distribution algorithm defined on the account's customer class to allocate the payment amongst the account's service agreements. In this section, we describe how to override this distribution (e.g., when you need to allocate a payment to specific service agreement(s)).

We start this explanation in the middle of [How To Add A New Payment Event](#).

In the **Payment Event Add** dialog, be sure to select **Do Not Distribute** before clicking **OK**.

- **Save** the payment event (this also saves the payment and tender information).
- Use the **Payment Amount** context menu to transfer to the **Manual Distribution** page.
- Fill in the desired **Distributed Amt** for each service agreement and save the payment.
- Click **Freeze** to freeze the payment when the amount distributed is equal to the payment amount.

How To Get An Unbalanced Tender Control In Balance (Fixing Over/Under)

In order to balance a tender control that is out-of-balance, you must create a tender for an account associated with your company.

The over / under account. Your organization must set up a company-use account with a service agreement whose SA type references the over/under distribution code. This account should be linked to the person ID associated with company usage.

- If you have more money in the drawer than you have tenders, then you're "over". In this situation, you need to record a payment for the "over" amount and the tender type configured for cash drawer over/under amounts. This will cause cash to be debited by the over amount, and the expense account associated with the account's service agreement to be credited.
- If you have less money in the drawer than you have tenders, then you're "under". In this situation, you need to record a negative payment for the "under" amount and the tender type configured for cash drawer over/under amounts. This will cause cash to be credited by the over amount, and the expense account associated with the account's service agreement to be debited.

Important! A separate tender type is used for recording cash drawer over/under amounts. Once the over/under amount has been determined, you must enter a new tender using your over/under tender type for the company-use account. Check out the following table for an example:

Tender Type	Starting Balance	Tenders Received	Turn Ins	Actual Ending Balance (Entered by Cashier)	Expected Ending Balance	Over/Under Amount
Cash	\$150.50	\$5,000	\$4,000	\$1151	\$1150.50	Over \$0.50
Check	-	\$1,000	\$750	\$249	\$250	Under \$1
Over/Under	-	\$-0.50	-	-	\$-0.50	Over \$0.50

In this example, you'd have to create a new tender for the company-use account for the amount that will be distributed to the company-use account's service agreement (debit over/under expense, credit cash):

- Tender type: Over/Under, Tender amount: -\$0.50

You must reopen the tender control. Before you can add an over/under tender to a tender control, you must **reopen** the tender control (tenders may only be added to **open** tender controls).

How To Add A Non-CIS Payment Using A Template

The payment portal allows you to add non-CIS payments using a payment template. Payment Templates can be configured for common types of payment allocations to pre-populate the payment distribution and allowing non-CIS payments to be directly allocated against distribution codes. The following steps explain how to do this.

You navigate through the pages as follows:

- From the main menu open **Financial, Payment Portal**.
- The **Add Non CIS Payment Zone** shows a list of all active payment templates. Use the **Add** button to add a non-CIS payment for a specific payment template.
- Enter the payment and tender details (**Payment Amount**, **Payment Date** defaults to the current date, **Tender Source** is used to find an open tender control when adding the tender).

Authorizing Tenders. If the **Tender Type** selected requires authorization (i.e. credit card payments), additional authorization details are required (e.g. credit card type, credit card number, etc.). Only one tender that requires authorization can exist for a single payment event.

For information about tender types requiring authorization, refer to [How To Authorize A Credit Card Tender](#).

- Enter the non-CIS information for the customer remitting the payment (**Customer Name**, **Agreement Number** and **Comments** to describe anything unusual about the non CIS payment).

Payor Account ID. The Non CIS Account Id specified on the payment template is used as the tendering account. If the account has exceeded your [cash-only threshold](#) and the **Tender Type** isn't marked as "like cash", a warning appears advising of such.

Including a CIS payment. If a portion of the tender is for a CIS payment, ensure that the **Add CIS Payment** checkbox is selected. The system will process the non-CIS payment, and then navigate the user to the Payment Event page where the CIS payment can be included.

- The distribution details are defaulted from the payment template but may be overridden. You can add additional distribution lines. Enter the payment amount and Reference Number for each applicable distribution line.
- When you click the **OK** button, the system creates the payment event and tender(s). A single payment is created for each non-zero amount entry in the distribution details list. The payment template is captured as a characteristic on the tender. The corresponding distribution code, GL account and reference number are captured as characteristics on each payment. When the financial transactions are created for these payments, the distribution code (and GL account) is retrieved from the payment characteristic collection.

How To Authorize A Credit Card Tender

You can use the Payment Portal to add a payment event for a tender that requires authorization. The following steps explain how to do this.

If your organization is using the base tender authorization functionality, you can navigate through the pages as follows:

- From the main menu open **Financial, Payment Portal**.
- The **Add CIS Payment Zone** only expands if there's an account in context. To expand this zone, either search for an account using Control Central or use the Tender Search available on the **Payment Portal**.
- Use the **Add Payment** button to add a payment for the account displayed in the zone.
- The information displayed is the same as that on the [Payment Event - Add Dialog](#), except for the following:
 - Use **Distribute Action** to describe what you'd like to have happen when you click the **OK** button. The dropdown contains the following values - ***Distribute and Freeze if OK***, ***Manual Distribution*** and ***Manual Distribution - Multiple Payments***

Note: The ***Distribute and Freeze if OK*** and ***Manual Distribution*** options are also present on the Distribute Action on the payment event add dialogue screen. The third option ***Manual Distribution - Multiple Payments*** is special for credit card authorization processing. This option is similar to the ***Manual Distribution*** option except that the user is navigated to the Payment Event maintenance page after authorization has taken place.

- If the **Tender Type** selected requires authorization (i.e. credit card payments), additional authorization details are required (e.g. credit card type, credit card number, etc). This information is displayed in the **Credit Card Information** section.

Authorizing Tenders. Only one tender that requires authorization can exist for a single payment event.

- The **Authorization Override** checkbox is only displayed if the tender type selected requires authorization. Use the **Authorization Override** checkbox to indicate whether or not tender authorization should take place. If **Authorization Override** is checked, an authorization code must be entered and the system bypasses the authorization web service call.
- Click **OK** to save the payment and navigate to the appropriate page.
- Assuming the credit card is accepted, the authorization code returned by the credit card authorization provider is captured as a tender characteristic.
- If the credit card is declined, the tender will not be added.

How To Reverse A Credit Card Tender

The steps required to reverse a credit card tender are the same as the steps for [How To Cancel A Tender](#).

The reversal code from the credit card authorization service is stored as a tender characteristic.

For more information on authorizing credit card payments, refer to the following business objects: ***Tender Type - Credit Card with Authorization, CyberSource - Credit Card Authorization and CyberSource - Credit Card Reversal.***

How To Process An Alternate Currency Payment

You can use the Payment Portal to add a payment event for a tender(s) in an alternate currency, i.e. a currency that differs from the account's currency. The following steps explain how to do this.

- From the main menu open **Financial, Payment Portal**.
- The **Add CIS Payment Zone** only expands if there's an account in context. To expand this zone, either search for an account using Control Central or use the Tender Search available on the **Payment Portal**.
- Use the **Add Payment** button to add a payment for the account displayed in the zone.
- The information displayed is the same as that on the [Payment Event - Add Dialog](#), except for the following:
 - If your organization accepts payment tenders in alternate currencies, the **Currency Converter** button appears in the tender list
- Use the **Currency Converter** button to enter the **Tendered Amount** and **Tendered Currency** in the alternate currency and click **OK**. The system will convert the alternate currency amount to the account's currency.
- Click **OK** to save the payment and navigate to the appropriate page.
- When you click the **OK** button, the system creates the payment event and tender(s). Each tender is linked to an appropriate tender control based on the currency of the tender.

Refer to [Alternate Currency Payments](#) for more information.

Financial Transactions On A Payment

Open **Financial Query, Financial Transactions On A Payment** to view the financial transactions on a payment.

Note. You can also open this page using the go to buttons that prefix the financial transaction summaries on [Payment - Main](#).

Description of page

Payment ID is the system-assigned unique identifier of the payment.

Account ID is the payment's account.

The area beneath **Account ID** provides you with options that control which financial transactions appear in the grid. The following points describe the various options:

- Use the **SA Filter** to define the types of service agreements whose financial transactions appear in the grid. The following options are available:
 - **Address.** Use this option to restrict financial transactions to those whose service agreements are linked to service points associated with a given **Address, City** and/or **Postal** code.
 - **All.** Use this option if you do not wish to restrict financial transactions based on service agreement attributes.
 - **Geographic Type.** Use this option to restrict financial transactions to those whose service agreements are linked to service points associated with a given **Geo Type** and **Value**.
 - **SA ID.** Use this option to restrict financial transactions to those of a specific service agreement.
 - **SA Type.** Use this option to restrict financial transactions to those whose service agreements are linked to a given **CIS Division** and **SA Type**.
- Use **Match Event Status Filter** to restrict the financial transactions based on the status of their [match event](#). This filter only appears if the payment's account is an [open-item](#) customer. The following options are available:
 - **All.** This option shows all financial transactions.
 - **Balanced.** This option shows all financial transactions whose match event is **balanced**.
 - **Disputed.** This option shows all financial transactions whose match event is **disputed**.
 - **Unbalanced.** This option shows all financial transactions whose match event is **unbalanced**.
 - **Unmatched.** This option shows all financial transactions that are not linked to a match event.

Don't forget to click the search button after changing the filters or after selecting a new Payment Id.

The grid that follows contains the financial transactions that match your search criteria. The following information is displayed:

- **Match Event Status** shows the status of the financial transaction's [match event](#). This column only appears if the account is an [open-item](#) customer.
- **FT Type** displays the type of financial transaction. Click on the hyperlink to transfer to [Financial Transaction - Main](#). On this page, you can change certain aspects of the FT in question.
- **Accounting Date** is the date the system uses to determine the financial transaction's accounting period in your general ledger.
- **Current Amount** contains the financial transaction's effect on the service agreement's current balance.
- **Payoff Amount** contains the financial transaction's effect on the service agreement's payoff balance. The **Payoff Amount** will be dim if it equals the **Current Amount**.
- **Show on Bill** indicates if information about the financial transaction appears on the customer's bill.
- **SA Information** contains a summary of the respective service agreement.
- **Financial Transaction ID** is the system-assigned unique identifier of the financial transaction.

At the bottom of the page is a summary of the financial transactions that match the search criteria.

Payment History

The payment history page shows all payments that have been distributed to an account's service agreements. Open this page using **Financial Query, Account Payment History**.

Description of Page

One row is displayed for every payment that has been distributed to an account's service agreements. The payment date, amount, payment status and the tender source associated with the tender's tender control are displayed for each payment.

Note. **Tender Source** typically contains the description of the cash drawer in which the payment was made or the remittance processor that processed the payment. **Tender source** will be blank for automatic payments until they are interfaced to the financial institution. Refer to [Downloading Automatic Payments](#) for more information about interfacing automatic payments.

If you need to see more detailed information about the payment, click the Go To button to transfer to the payment event page.

Payment Tender History

The **Payment Tender History** page shows a list of all tenders linked to an open tender control for the current user. Open this page using **Financial Query, Payment Tender History**.

Description of Page

One row is displayed for every tender that is linked to an open tender control to which the current user has access. The freeze date/time, payor account id, name, tender amount, tender type, pay tender id, tender status and payment date are displayed for each tender.

Payment / Tender Search

This page allows you to look for payments and/or tenders using a combination of search criteria. Open this page using **Financial Query, Payment / Tender Search**.

Description of Page

The top half of the page is where you enter the criteria used to search for payments and tenders.

Multiple search criteria may be specified. You can search for payments and tenders using a combination of search criteria. For example, if you enter both a **Payment Account** name of **Brandon** and a **Payment Amount** between **150** and **170**, only those payments for customers named **Brandon** whose amount is between **150** and **170** will be displayed.

Warning! Try to be as specific as possible when entering search criteria. Why? Because entering open-ended search criteria may have a severe impact on response times. For example, if you know the payments you're looking for have a payment date sometime in **January 2003** and the customer's name is **Brazil,John**, then enter both of these criteria. If you also know the payment amount is between **150** and **170**, enter these values too.

The following table describes each of the different search methods.

Search Method	Description
Search for	Use this field to define if you're searching for <i>Payments</i> , <i>Tenders</i> or both <i>Payments and Tenders</i> . The value entered in this field controls which of the remaining search methods are enabled.
Distribution Rule	Use this field if you're searching for a tender or a payment and know the distribution rule and value used to created it. Note. This section appears only if you have configured your system to allow the payment event distribution rules method to be used.
Payment Account	Use this field if you're searching for a payment and know the account ID or the name of any person linked to the account whose debt is relieved by the payment. - If you know the payment's account ID, first choose <i>Account</i> in the adjacent dropdown and then enter the account's ID. You must enter all of the account ID. - If you know the name of any person linked to the account, first choose <i>Person Name</i> in the adjacent dropdown and then enter the person's name. You can enter all or part of the person's name (the more you enter, the faster the search will be). The name

Search Method	Description
	<p>search is not case sensitive.</p> <p>A search method of <i>Person Name</i> defaults.</p> <p>If you leave the person name or ID blank, the system ignores this search method.</p> <p>These fields are protected if you've selected a Search for of <i>Tender</i>.</p>
Payment Amount	<p>Use this field if you're searching for a payment and know its amount.</p> <ul style="list-style-type: none"> - If you know the exact amount, first choose <i>Equal To</i> in the adjacent dropdown and then enter the amount. - If you know the payment amount is between a given range of values, first choose <i>Between</i> and then enter the range of values. <p>A search method of <i>Between</i> defaults.</p> <p>If you leave the amount(s) blank, the system ignores this search method.</p> <p>These fields are protected if you've selected a Search for of <i>Tender</i>.</p>
Payor Account	<p>Use this field if you're searching for a tender and know the account ID or the name of any person linked to the tendering account.</p> <ul style="list-style-type: none"> - If you know the tender's account ID, first choose <i>Account</i> in the adjacent dropdown and then enter the account's ID. You must enter all of the account ID. - If you know the name of any person linked to the tendering account, first choose <i>Person Name</i> in the adjacent dropdown and then enter the person's name. You can enter all or part of the person's name (the more you enter, the faster the search will be). The name search is not case sensitive. <p>A search method of <i>Person Name</i> defaults.</p> <p>If you leave the person name or ID blank, the system ignores this search method.</p> <p>These fields are protected if you've selected a Search for of <i>Payment</i>.</p>
Tender Amount	<p>Use this field if you're searching for a tender and know its amount.</p> <ul style="list-style-type: none"> - If you know the exact amount, first choose <i>Equal To</i> in the adjacent dropdown and then enter the amount. - If you know the tender amount is between a given range of values, first choose <i>Between</i> and then enter the range of values. <p>A search method of <i>Between</i> defaults.</p> <p>If you leave the amount(s) blank, the system ignores this search method.</p> <p>These fields are protected if you've selected a Search for of <i>Payment</i>.</p>
Tender Source	<p>If you're searching for a tender and you know the source of the tender (e.g., the lockbox, cashier, etc.), enter the tender source code.</p> <p>A search method of <i>Equal To</i> defaults.</p> <p>If you leave the tender source blank, the system ignores this search method.</p> <p>These fields are protected if you've selected a Search for of <i>Payment</i>.</p>
MICR ID	<p>Use this field if you're searching for a tender and know its MICR (magnetic ink character recognition) ID.</p> <ul style="list-style-type: none"> - If you know the entire MICR ID, first choose <i>Equal To</i> in the adjacent dropdown and then enter the entire MICR ID.

Search Method	Description
	<p>- If you know the first X characters of the MICR ID, first choose <i>Like</i> in the adjacent dropdown and then enter the entire MICR ID.</p> <p>A search method of <i>Like</i> defaults.</p> <p>If you leave the MICR ID blank, the system ignores this search method.</p> <p>These fields are protected if you've selected a Search for of <i>Payment</i>.</p>
Payment Date	<p>Use this field if you're searching for a tender or a payment and know its date.</p> <p>- If you know the exact date, first choose <i>Equal To</i> in the adjacent dropdown and then enter the date.</p> <p>- If you know the date is between a given range of values, first choose <i>Between</i> and then enter the date range.</p> <p>A search method of <i>Between</i> defaults.</p> <p>If you leave the date(s) blank, the system ignores this search method.</p>

The system shows the total number of payments / tenders that satisfy your search results immediately below the grid.

The first group of payments / tenders is displayed in the grid at the bottom of the page. Different columns appear in the grid depending on the value of **Search for**.

- If you use a **Search for** of *Payment*, only payment-oriented columns appear in the grid.
- If you use a **Search for** of *Tender*, only tender-oriented columns appear in the grid.
- If you use a **Search for** of *Payment and Tender*, both payment-oriented and tender-oriented columns appear in the grid.

Warning! If you use the *Payment and Tender* search method, and the resultant tenders / payments are linked to payment events that have multiple payments or tenders, multiple rows may be displayed for the payment event's tenders and payments. For example, if a tender with multiple payments is selected, a separate row will be displayed for every payment that matches your payment search criteria. If you don't specify any payment search criteria, a row will be displayed for every payment linked to the tender.

Payment Date	This contains the date of the payment event associated with the payment / tender. This column appears regardless of the value of Search for .
Payment Account Info	This contains a concatenation of important information about the account whose debt was relieved by the payment. This column does not appear if you use a Search for of <i>Tender</i> .
Payment Amount	This contains the amount of the payment. This column does not appear if you use a Search for of <i>Tender</i> .
Payment Status	This contains the status of the payment. Refer to Payment Lifecycle for more information. This column does not appear if you use a Search for of <i>Tender</i> .

Payment ID	This contains the unique identifier of the payment. This column does not appear if you use a Search for of <i>Tender</i> .
Payor Account Info	This contains a concatenation of important information about the account who tendered the payment. This column does not appear if you use a Search for of <i>Payment</i> .
Tender Amount	This contains the amount of the tender. This column does not appear if you use a Search for of <i>Payment</i> .
Tender Source	This contains the source of the tender of the tender. This column does not appear if you use a Search for of <i>Payment</i> .
MICR ID	This contains the MICR (magnetic ink character recognition) ID of the tender. This column does not appear if you use a Search for of <i>Payment</i> .
Tender Status	This contains the status of the tender. Refer to Tender Lifecycle for more information. This column does not appear if you use a Search for of <i>Payment</i> .
Pay Tender ID	This contains the unique identifier of the tender. This column does not appear if you use a Search for of <i>Payment</i> .
Pay Event ID	This contains the unique identifier of the payment event.

Payment Event Exception

Unbalanced payment events cause a record to be written to the payment event exception table with a message indicating the nature of the error.

To view the messages associated with the exception records, schedule the [TD-UNBAL](#) background process. This process generates a To Do entry for every record in the payment event exception table.

Refer to [Unbalanced Payment Events](#) for instructions describing how to correct an unbalanced payment event.

Payment Exception

When the system is unable to distribute a payment, a record to be written to the payment exception table with a message indicating the nature of the error.

To view the messages associated with the exception records, schedule the [TD-PYERR](#) background process. This process generates a To Do entry for every record in the payment exception table.

After correcting the cause of the error, drill into the [Payment](#) and attempt to redistribute it.

Maintaining Deposit Controls

Deposit controls exist to give you administrative control over your cash drawers (and all other tender sources) and the subsequent deposit of funds at banks.

Refer to [Tender Management and Workstation Cashiering](#) for background information.

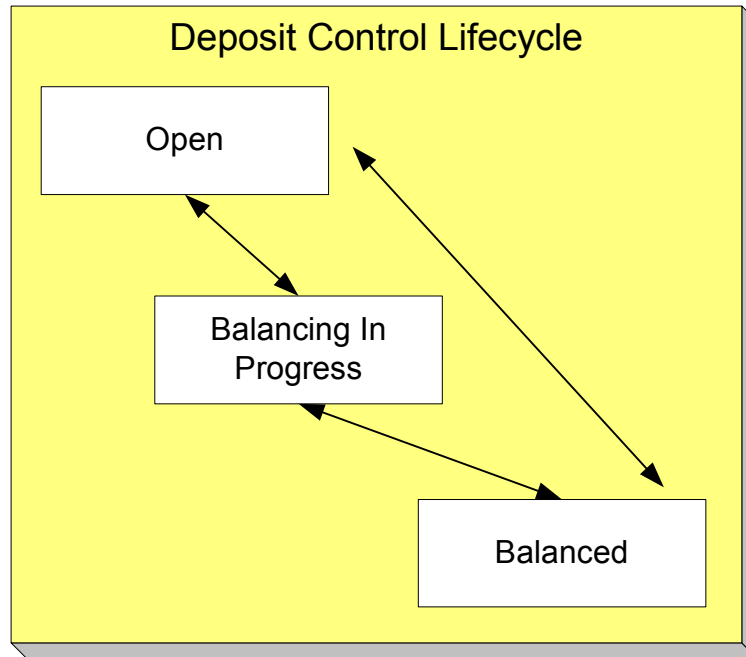
The system creates most deposit controls behind-the-scenes. Most deposit controls are created by the system when it processes tenders from your remittance processor and lock boxes. You should only have to access the deposit control pages if you record payments in cash drawers. For information about how the system creates deposit controls, refer to [Managing Payments Interfaced From External Sources](#). Also note that the automatic payment activation process also creates tender and deposit controls. Refer to [Activating Automatic Payments](#) for more information.

Contents

- [The Lifecycle Of A Deposit Control](#)
- [Deposit Control - Main](#)
- [Deposit Control - Tender Control](#)
- [Deposit Control - Tender Deposit](#)
- [Deposit Control - Turn Ins](#)
- [Deposit Control - Characteristics](#)

The Lifecycle Of A Deposit Control

The following diagram shows the possible lifecycle of a deposit control.

**Open**

A deposit control is initially created in the **Open** state. While in this state, you may add new deposits to it, change the deposit amount on its deposits, and transfer tender controls into and out of it.

Balancing In Progress

You change a deposit control's status to **Balancing In Progress** when you're ready to balance its contents. While in this state, you can change the deposit amount on its deposits and transfer tender controls out of it. If you need to add new deposits to it or transfer tender controls into it you must return it to the **Open** state.

Balanced

You change a deposit control's status to **Balanced** when the sum of its tender controls is consistent with the total of its deposits. While in this state, you cannot modify its deposits or its tender controls. If you need to make modifications, you must return it to the **Open** or **Balancing In Progress** state.

Background processes and state transition. When payments are interfaced from external sources, the system automatically creates a deposit control and links a tender control to it. When all payments have been successfully loaded, the system changes the state of the respective deposit control to **Balanced**.

Deposit Control - Main

The Main page contains core deposit control information. Open this page using **Financial, Deposit Control** and then navigate to the **Main** tab.

Description of Page

Deposit Control contains a concatenation of the deposit control's creation date, tender source type, and status.

Deposit Control ID is the system-assigned, unique identifier of the deposit control.

User is the person who created the deposit control.

Create Date/Time is the date and time the deposit control was created.

Tender Source Type is the type of tender control that has been linked to the deposit control. Valid values are: *Ad hoc*, *Auto Pay*, *Online Cashiering* and *Lockbox*. The system uses this information to prevent tender controls from different sources from being included under the same deposit control. In other words, you can't mix automatic payment, cashiering and lockbox tenders under the same deposit control.

Currency Code is the currency in which the deposit control's tenders are denominated.

Default note. The currency code defaults from the installation record.

The summary information that follows contains a summary of the starting balance and the tenders that are linked to the tender control.

- **Starting Balance** is the sum of the starting balances from all tender controls linked to the deposit control.
- **Total Tenders Amount** is the sum of tenders from all tender controls linked to the deposit control.
- **Total Tender Controls** is the number and amount of tender controls linked to the deposit control.
- **Total Tender Deposits** is the number and amount of tender deposits linked to the deposit control.
- **Expected Ending Balance** is the **Total Tender Control** minus **Total Tender Deposits**.
- **Ending Balance** is the actual amount of money in the tender control. This amount must equal the **Expected Ending Balance** before the tender control can be marked as *Balanced*.
- **Outstanding Over/Under** is **Ending Balance** minus **Expected Ending Balance**.

Deposit Control Status shows the status of the deposit control. Valid values are *Open*, *Balanced*, and *Balancing In Progress*.

For more information, refer to [The Lifecycle Of A Deposit Control](#).

The **Balanced User ID** and **Balanced Date/Time** are populated when the status is changed to *Balanced*.

Use **Comments** to describing anything unusual about the deposit control.

Deposit Control - Tender Control

The Tender Control page contains a row for every tender control linked to the deposit control. Open this page using **Financial, Deposit Control, Tender Control**.

Description of Page

Deposit Control contains a concatenation of the deposit control's creation date, tender source type, and status.

Deposit Control ID is the system-assigned, unique identifier of the deposit control.

The grid that follows contains a row for every tender control linked to the deposit control. No information about the tender controls may be modified on this page. To view or modify tender control information, click the Go To button.

Deposit Control - Tender Deposit

The Tender Deposit tab page contains a row for every tender deposit linked to the deposit control. Open this page using **Financial, Deposit Control** and navigate to the **Tender Deposit** tab.

Description of Page

Deposit Control contains a concatenation of the deposit control's creation date, tender source type, and status.

Deposit Control ID is the system-assigned, unique identifier of the deposit control.

The **Tender Deposit** scroll that follows contains one row for every financial institution into which the tenders will be deposited. To insert a new row, click the + button and fill in the following fields:

- **Tender Deposit ID** is maintained by the system. When a deposit is being created, there is no ID number displayed. Once a deposit is entered and saved, the system generates an ID and displays it here.
- **Deposit Amount** contains the amount to be deposited at the bank. **Currency Code** is the currency in which the deposit is denominated.
- **Reference ID** contains the ID of the deposit (if any).
- Use **Bank Code** and **Bank Account** to define where the tenders will be deposited.

Deposit Control - Turn Ins

The Turn Ins page contains a row for every turn-in event linked to the deposit control. Open this page using **Financial, Deposit Control** and navigate to the **Turn Ins** tab.

Refer to [Turn Ins](#) for background information.

Description of Page

Deposit Control contains a concatenation of the deposit control's creation date, tender source type, and status.

Deposit Control ID is the system-assigned, unique identifier of the deposit control.

The turn-ins grid contains one row for every turn-in event associated with the deposit control. A turn-in event is created when moneys originally deposited in respect of a tender control are turned in to the tender control's deposit control. Turn-in events are created and maintained on the [Tender Control – Turn Ins](#) page. Refer to [Turn Ins](#) for background information. The following information is displayed in the grid:

- **Turn In Status** defines if it is a new turn in *Awaiting approval* or has been *Approved* by the operator who is responsible for the Deposit Control. Once the turn-in is *Approved*, this field becomes display-only.
- **Tender Type** is the type of tender that has been turned in. This is a display-only field.
- **Turn In Amount** is the amount of the **Tender Type** that has been turned in. This is a display-only field.
- **Receipt Number** is the ID of the receipt given to the person who turn-in the funds. This is a display-only field.
- **Create Date/Time** contains when the turn-in event was created. This is a display-only field.

Deposit Control - Characteristics

To update deposit control characteristics, open **Financial, Deposit Control** and navigate to the **Characteristics** tab.

Description of Page

Deposit Control contains a concatenation of the deposit control's creation date, tender source type, and status.

Deposit Control ID is the system-assigned, unique identifier of the deposit control.

The **Characteristics** collection contains information that describes miscellaneous information about the deposit control.

The following fields display:

Characteristic Type	Indicate the type of characteristic.
Sequence	Controls the order in which characteristics of the same type are displayed.
Characteristic Value	Indicate the value of the characteristic.

Maintaining Tender Controls

Tender controls exist to give you administrative control over your cash drawers (and all other tender sources).

Refer to [Managing Your Cash Drawers](#) for more information.

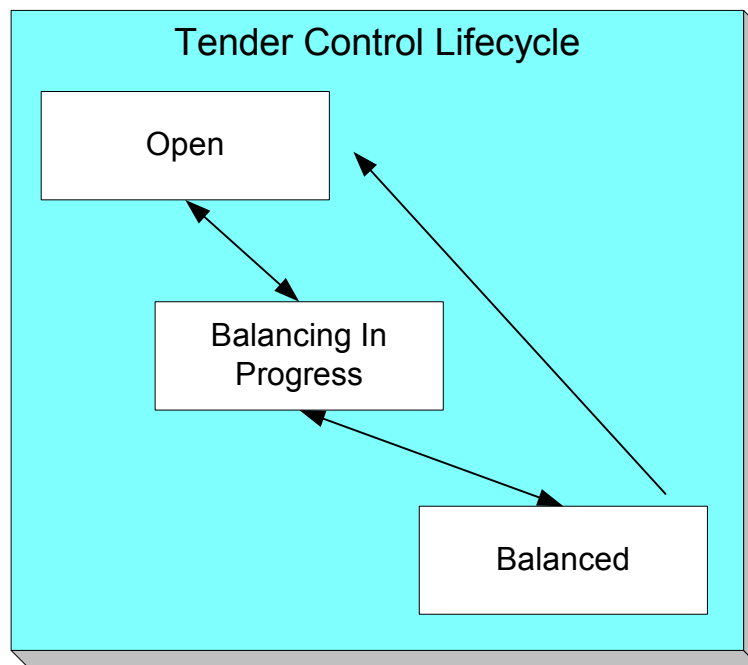
The system creates most tender controls behind-the-scenes. Most tender controls are created by the system when it processes tenders from the remittance processors and lock boxes. You should only have to access the tender control pages if you record payments in cash drawers. For information about how the system creates tender controls, refer to [Managing Payments Interfaced From External Sources](#). Also note that the automatic payment activation process also creates tender and deposit controls. Refer to [Activating Automatic Payments](#) for more information.

Contents

- [The Lifecycle Of A Tender Control](#)
- [Tender Control - Main](#)
- [Tender Control - Tenders](#)
- [Tender Control - Turn Ins](#)
- [Tender Control - Exceptions](#)
- [Tender Control - Characteristics](#)

The Lifecycle Of A Tender Control

The following diagram shows the possible lifecycle of a tender control.



Open

A tender control is initially created in the **Open** state. While in this state, you may add new tenders to it, change the tender amount on its tenders, and transfer tenders into and out of it.

Balancing In Progress

You change a tender control's status to **Balancing In Progress** when you're ready to balance its contents. While in this state, you can change the tender amount on its tenders and transfer tenders out of it. If you need to add new tenders to it or transfer tenders into it, you must return it to the **Open** state.

Balanced

You change a tender control's status to **Balanced** when the sum of its tenders is consistent with the ending balance in the drawer. While in this state, you cannot modify it or its tenders. If you need to make modifications, you must return it to the **Open** or **Balancing In Progress** state.

If the tender control is part of a **Balanced** deposit control, you may not change its status.

Background processes and state transition. When payments are interfaced from external sources, the system automatically creates a tender control and links tenders to it (one for each payment interfaced). When all payments have been successfully loaded, the system changes the state of the respective tender control to **Balanced**.

Tender Control - Main

The Main page contains core tender control information. This page is used to balance the tender control.

Open this page using **Financial, Tender Control**.

Searching For Tender Controls. When you use the **Creation Date** to search for tender controls, the system returns tender controls that are created on or before the date you specify. If the number of records returned exceeds the search limit (i.e. 300 records), only the records that do not exceed search limit are displayed. Therefore, you may need to specify an earlier creation date to find the record you are looking for. If you do not specify a creation date and search based on other criteria, the system uses the most recent date.

Refer to [Managing Your Cash Drawers, Turn Ins](#) and [Balancing By Tender Type](#) for more information.

Description of Page

Tender Control contains a concatenation of the tender control's creation date, tender source, and status.

Tender Control ID is the system-assigned, unique identifier of the tender control.

Tender Source is the source of the tenders (e.g., cash drawer 22, lockbox 1 at Bank of America).

Deposit Control ID is the system-assigned, unique identifier of the deposit control that the tender control is part of. The deposit control's **Currency Code** is displayed below.

For more information about deposit controls and tender controls, refer to [Managing Your Cash Drawers](#).

Turn on **All Users** if any user may insert tenders into this tender control. Turn this switch off and specify the appropriate **User ID** if only a single user may insert tenders into this tender control.

Create Date/Time is the date and time the tender control was created.

The **Starting Balance** of the tender control is defaulted from the tender source. This may be overridden while the **Tender Control Status** is **Open** or **Balancing in Progress**.

The **Tender Control Status** shows the status of the tender control. The **Balanced User ID** and **Balanced Date/Time** are populated when the status is changed to **Balanced**.

For more information, refer to [The Lifecycle Of A Tender Control](#).

The **Balance** button is enabled when the **Tender Control Status** is **Balancing in Progress**. When this button is clicked, the system checks the following:

- All Turn-Ins have been **Approved**.
- The sum of all **Ending Balances** equals the sum of all **Expected Ending Balances**.

If the above conditions are true, the status of the tender control is changed to **Balanced** and all modifiable fields become protected.

The tenders grid contains a summary of the tenders linked to the tender control. One row is displayed for each tender type. Each row contains the number of tenders of this type and their total amount. In order to **Balance** the tender control, you must enter an appropriate **Ending Balance** for each **Tender Type**. Refer to [Balancing By Tender Type](#) for more information. The following information appears in the grid:

- **Tender Type** represents the type of tender (e.g., cash, credit card). This is a display-only field.
- **Nbr of Tenders** contains the total number of tenders of this type in the tender control. This is a display-only field and is calculated by the system by accumulating the tenders linked to the tender control.
- **Total Tenders** contains the total amount of tenders of this type in the tender control. This is a display-only field and is calculated by the system by accumulating the tenders linked to the tender control. Navigate to the **Tenders** page to see the individual tenders.
- **Turn In Amount** contains the total amount of turn-ins of this type. This is a display-only field and is calculated by the system by accumulating the turn-ins linked to the tender control. Navigate to the **Turn Ins** pages to see the individual turn ins.
- **Starting Balance** contains the starting balance of this type of turn in. This field is only populated on the row defined as the Starting Balance tender type on the Installation Record. This is a display-only field and is equal to the tender control's **Starting Balance**.
- **Expected Ending Bal** is the **Starting Balance** plus **Total Tenders** minus **Turn In Amount** for this type of tender.

- **Ending Bal** is the actual amount of money of this tender type. The cashier must enter this field in order to balance the tender control. This field is protected unless the **Status** of the tender control is **Balancing In Progress**. This amount must equal the Expected Ending Balance before the tender control can be marked as **Balanced**. Refer to [How To Get An Unbalanced Tender Control In Balance \(Fixing Over/Under\)](#) for more information.
- **Outstanding Over/Under** is the difference between **Expected Ending Bal** and **Ending Bal**. This is a display-only field.

The summary information beneath the tender type grid contains a summary of the tenders that are linked to the tender control.

- **Total Tender Amount** is the total amount of all tenders linked to the tender control.
- **Ending Balance** is the actual amount of money in the tender control (as defined in the above grid).
- **Expected Ending Balance** is the expected amount of money in the tender control (as defined in the above grid).
- **Outstanding Over/Under** is Ending Balance minus Expected Ending Balance. If this amount is zero, the field is not displayed.

If the tender controls were created for a background process, such as the automatic payment extract, the following information is displayed:

- **Batch Code** is the batch code for which the tender control was created. This field only appears if it contains a value.
- **Batch Number** is the batch number for which the tender control was created. This field only appears if it contains a value.

Use **Comments** to describe anything unusual about the tender control (e.g., to explain why a large over/under amount was created).

Tender Control - Tenders

The Tenders page contains a row for every tender linked to the tender control. Open this page using **Financial, Tender Control, Tenders**.

Description of Page

Tender Control contains a concatenation of the tender control's creation date, tender source, and status.

Tender Control ID is the system-assigned, unique identifier of the tender control.

This grid at the top of the page contains a row for every tender linked to the tender control. When first displayed, the grid is ordered by the Create Date/Time column. No information about the tenders may be modified. To view or modify tender information, click the Go To button.

The grid at the bottom of the page contains a summary by tender type of all tenders linked to the tender control.

Tender Control - Turn Ins

Every time you turn in funds to the deposit control that is associated with your tender control, you create a turn-in event using this page. Open this page using **Financial, Tender Control** and navigate to the **Turn Ins** tab.

Refer to [Managing Your Cash Drawers](#) and [Turn Ins](#) for more information.

Description of Page

Tender Control contains a concatenation of the tender control's creation date, tender source, and status.

Tender Control ID is the system-assigned, unique identifier of the tender control.

The turn-ins grid contains one row for every turn-in event associated with the tender control. A turn-in event is created when moneys are physically transferred to the deposit control associated with the tender control. Turn-in events are approved on the [Deposit Control – Turn Ins](#) page. The following information is displayed in the grid:

- **Turn In Status** defines if the turn-in is **Awaiting Approval** or has been **Approved** by the operator who is responsible for the Deposit Control. When a turn-in is first created, its status is **Awaiting Approval** and the following fields should be defined. Once the turn-in is **Approved** by the user responsible for the deposit control associated with the tender control, all of the following fields become display-only and the turn-in event cannot be deleted.
- **Tender Type** is the type of tender that has been turned in. This field may only be modified when the **Turn In Status** is **Awaiting Approval**.
- **Turn In Amount** is the amount of the **Tender Type** that has been turned in. This field may only be modified when the **Turn In Status** is **Awaiting Approval**.
- **Receipt Number** is the ID of the receipt given to the person who turn-in the funds. This field may only be modified when the **Turn In Status** is **Awaiting Approval**.
- **Create Date/Time** contains when the turn-in event was created. This is a display-only field.

Tender Control - Exceptions

This page contains an entry for every exception (i.e., error) associated with the tender control's payments and payment events. Open this page using **Financial, Tender Control** and navigate to the **Exceptions** tab.

Refer to [Exceptions](#) for more information.

Description of Page

This page contains an entry for every exception (i.e., error) associated with the tender control's payments and payment events.

Tender Control - Characteristics

To update tender control characteristics, open **Financial, Tender Control** and navigate to the **Characteristics** tab.

Note: If your tender control is balanced, you cannot update tender control characteristics.

Description of Page

The **Characteristics** collection contains information that describes miscellaneous information about the tender control.

The following fields display:

Characteristic Type	Indicate the type of characteristic.
Sequence	Controls the order in which characteristics of the same type are displayed.
Characteristic Value	Indicate the value of the characteristic.

Interfacing Payments From External Sources

Most payments are NOT added by an operator using the Payment Event page. Rather, they are interfaced from an external source (e.g., a lock box or a remittance processor).

The base-package provides two interfaces to upload payments, each based on a different method of creating payment events.

Refer to [Distributing A Payment Event](#) for more information about how payment event distribution is handled in the system.

The topics in this section describe how these payment interfaces work.

Contents

- [Interfacing Payments](#)
- [Interfacing Payments Using Distribution Rules](#)

Interfacing Payments

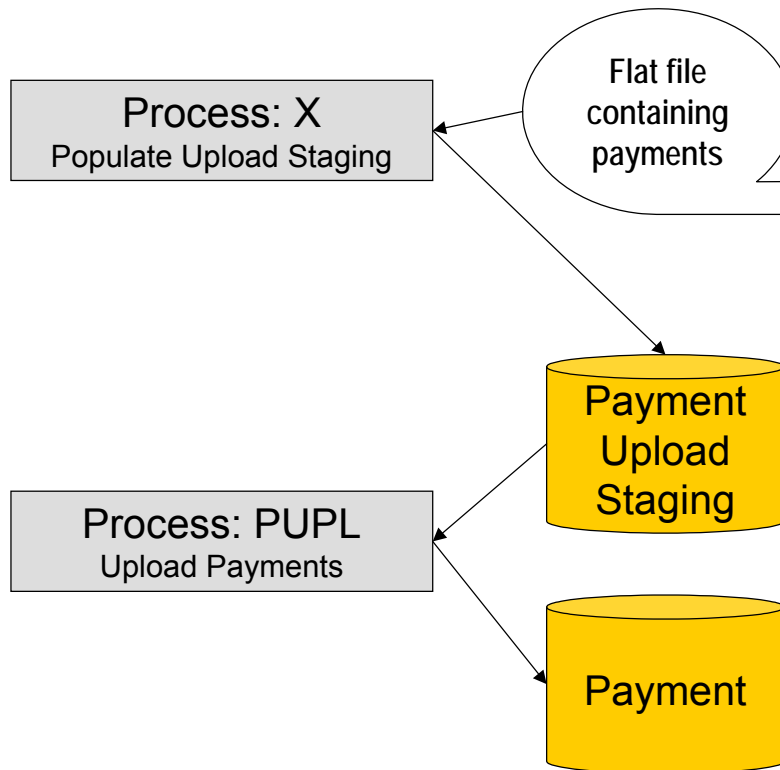
The topics in this section describe how the payment interface using the system default method of creating payment events works.

Contents

- [Populating The Payment Upload Staging Records](#)
- [PYUP-PRG - Purge Payment Upload Objects](#)
- [Maintaining Deposit Control Staging](#)
- [Payment Upload Staging](#)
- [Payment Upload Exception](#)

Populating The Payment Upload Staging Records

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



The topics in this section describe each background process referenced above.

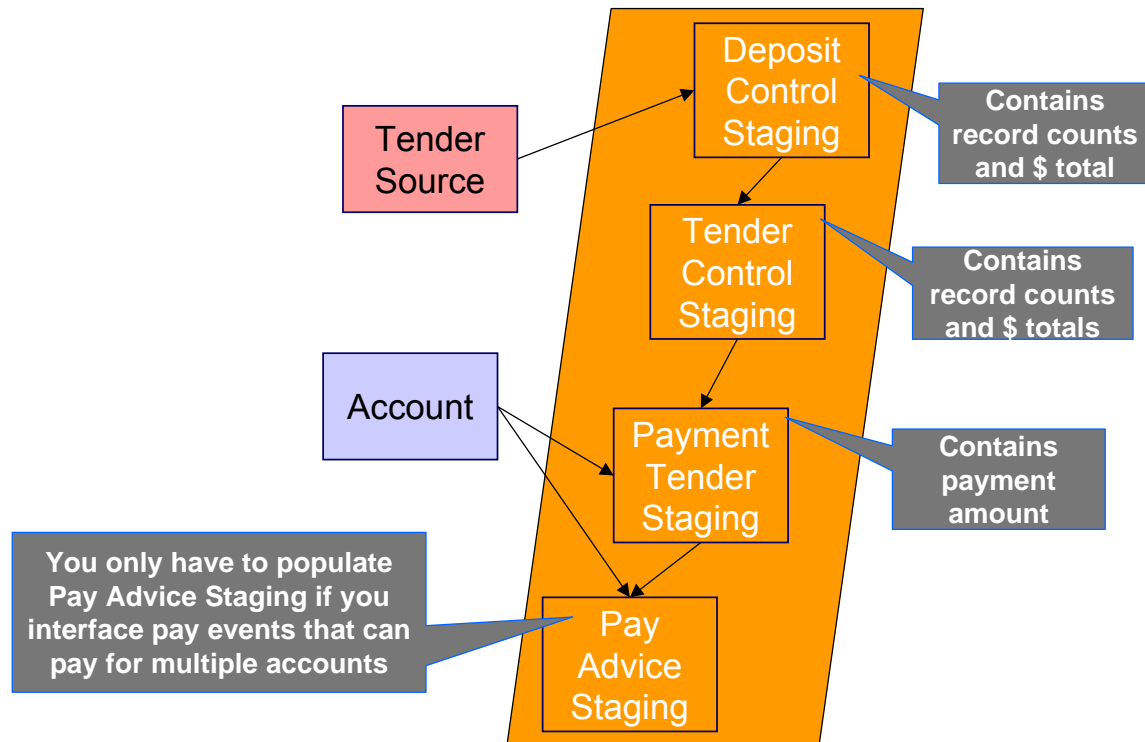
Contents

[Process X - Populate Payment Upload Staging](#)

[Process PUPL - Upload Payments](#)

Process X - Populate Payment Upload Staging

Process X refers to the mechanism used by your organization to populate the various staging tables (shown in the orange section of the following ERD).



The topics in this section describe each of these tables.

Contents

[Deposit Control Staging](#)
[Tender Control Staging](#)
[Payment Tender Staging](#)
[Payment Advice Staging](#)

Deposit Control Staging

You must create a deposit control staging record for each batch of payments to be uploaded into the system. The name of this table is [CI DEP CTL ST](#). The following table describes each column on this table.

Column Name	Length	Req'd	Data Type	Comments
EXT_SOURCE_ID	30	Y	A/N	This must correspond with an external source ID on one of the defined tender sources. Refer to Setting Up Tender Sources for more information.
EXT_TRANSMIT_ID	30	Y	A/N	This is the unique identifier of the transmission from the external source. This must be a unique value for each transmission from the source.
DEP_CTL_STG_ST_FLG	2	Y	A/N	This must be set to 20 (20 is the lookup value that corresponds with the <i>Pending</i> state)
DEP_CTL_ID	10	N	N	Leave this column blank. It will be assigned by the system when it creates a deposit control record.
TRANSMIT_DTTM	15	Y	DateTi	Date and time that the file was transmitted.

			me	
CURRENCY_CD	3	Y	A/N	This must be a valid currency code (this would be <i>USD</i> for United States dollars).
TOT_TNDR_CTL_AMT	13.2	Y	N	This column must equal the sum of the payment amounts on the tender control staging records associated with this deposit control staging.
TOT_TNDR_CTL_CNT	10	Y	N	This column must equal the number of tender control staging records associated with this deposit control staging.
LAST_UPDATE_INST	10	N	N	This field is populated during the upload. It is the process scheduler instance ID of the process performing the upload.

You must create one or more [Tender Control Staging](#) for this deposit control staging record.

Tender Control Staging

You must create at least one tender control staging record for each batch of payments to be uploaded into the system. The name of this table is [CI_TNDR_CTL_ST](#). The following table describes each column on this table.

Column Name	Length	Req'd	Data Type	Comments
EXT_SOURCE_ID	30	Y	A/N	This must correspond with the external source ID on the parent deposit control staging record.
EXT_TRANSMIT_ID	30	Y	A/N	This must correspond with the external transmission ID on the parent deposit control staging record.
EXT_BATCH_ID	30	Y	A/N	This is the unique identifier of the batch of payments in respect of the external transmission ID.
TNDR_CTL_STG_ST_FLG	2	Y	A/N	This must be set to 20 (20 is the translate value that corresponds with the <i>Pending</i> state)
TNDR_CTL_ID	10	N	N	Leave this column blank. It will be assigned by the system when it creates a tender control record.
TOT_TNDR_AMT	13.2	Y	N	This column must equal the sum of the payment amounts on the payment tender staging records associated with this tender control staging.
TOT_TNDR_CNT	10	Y	N	This column must equal the number of payment tender staging records associated with this tender control staging.

You must create one or more [Payment Tender Staging](#) records for this tender control staging record.

Payment Tender Staging

You must create at least one payment tender staging record for each payment associated with the tender control staging record. The name of this table is [CI_PAY_TNDR_ST](#). The following table describes each column on this table.

Column Name	Length	Req'd	Data Type	Comments
EXT_SOURCE_ID	30	Y	A/N	This must correspond with the external source ID on the parent deposit control staging record.
EXT_TRANSMIT_ID	30	Y	A/N	This must correspond with the external transmission ID on the parent tender control staging record.
EXT_BATCH_ID	30	Y	A/N	This must correspond with the external batch ID on the parent tender control staging record.
EXT_REFERENCE_ID	30	Y	A/N	This is the unique identifier of the payment in respect of the external batch ID.
PAY_TND_STG_ST_FLG	2	Y	A/N	This must be set to 10 (10 is the translate value that corresponds with the <i>Pending</i> state)
PAY_TENDER_ID	12	N	N	Leave this column blank. It will be assigned by the system when it creates a tender record.
TENDER_AMT	13.2	Y	N	The amount tendered (i.e., the payment amount).
ACCOUNTING_DT	10	Y	Date	This is the date that should be used for accounting purposes. This should correspond with an open accounting period.
TENDER_TYPE_CD	4	Y	A/N	This must correspond with the prime key of one of your tender types. Refer to Setting Up Tender Types for more information.
CUST_ID	15	Y	A/N	This is the account ID or old account number of the customer tendering the payment. If the system cannot find an account ID or old account number that matches this value, the account ID of the tender source's suspense SA will be used on the corresponding tender and payment.
MICR_ID	30	N	A/N	This is the MICR ID associated with the payment.
NAME1	40	N	A/N	This is the customer name on the payment.
CHECK_NBR	10	N	A/N	This is the check number on the payment.

Payment Advice Staging

You need only populate rows on this table if any of the following conditions apply:

- If you need to distribute a payment tender to an account other than that defined with the CUST_ID on the payment tender staging record, you must create a payment staging record. You may distribute a tender to multiple accounts by creating multiple payment staging records. Note, if you want to distribute the payment tender to the same account, you do NOT need a payment staging record.
- If you want to restrict a payment to a specific service agreement, you must insert a row on this table to indicate the specific the service agreement in question. You do this by populating MATCH_TYPE_CD with a value that indicates that you are paying for a specific service agreement and MATCH_VALUE with the unique ID of the service agreement.

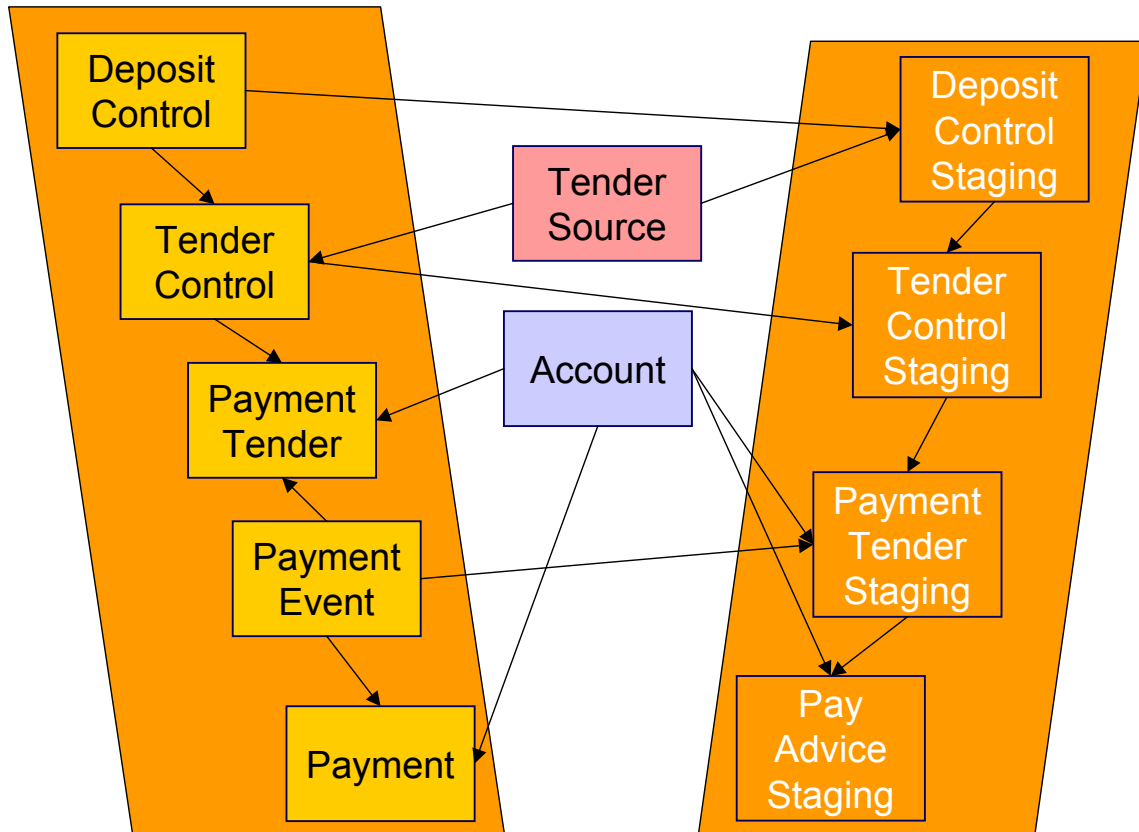
- If you practice [open-item accounting](#), you must insert a row on this table for each to indicate the open-item to which the payment should be matched. Note, because open-item customer typically match payments to bills, you would populate MATCH_TYPE_CD with a value to indicate that you are matching by bill ID and MATCH_VALUE with the unique ID of the bill.

The name of this table is [CI_PAY_ST](#). The following table describes each column on this table.

Column Name	Length	Req'd	Data Type	Comments
EXT_SOURCE_ID	30	Y	A/N	This must correspond with the external source ID on the parent payment tender staging record.
EXT_TRANSMIT_ID	30	Y	A/N	This must correspond with the external transmission ID on the parent payment tender staging record.
EXT_BATCH_ID	30	Y	A/N	This must correspond with the external batch ID on the parent payment tender staging record.
EXT_REFERENCE_ID	30	Y	A/N	This must correspond with the external reference ID on the parent payment tender staging record.
CUST_ID	15	Y	A/N	This is the account ID or old account number of the customer to which the payment should be distributed. If the system cannot find an account ID or old account number that matches this value, the account ID of the payor is used on the corresponding payment. If the payor's account ID is invalid, the tender source's suspense SA is used.
PAY_AMT	13.2	Y	N	The amount tendered (i.e., the payment amount).
MATCH_TYPE_CD	8	N	A/N	See the description of the MATCH_VALUE field below. Refer to Payments And Match Events for more information about the significance of this field.
MATCH_VALUE	30	N	A/N	MATCH_VALUE and MATCH_TYPE_CD are used in conjunction to indicate that the distribution of the payment should be restricted in some way (i.e., the standard payment distribution algorithm should not be used). MATCH_TYPE_CD indicates how the payment should be distributed (e.g., only distribute to a specific service agreement), MATCH_VALUE contains the ID of the restriction (e.g., the SA ID). If MATCH_TYPE_CD is specified, it must reference a valid Match Type .

Process PUPL - Upload Payments

The batch process identified by the batch process ID **PUPL** refers to the background process that loads the contents of the various payment staging records into the various payment event tables. The tables that are populated by this process are shown in the left orange section of the following ERD (the right orange section are the staging tables populated by the process described above)



The topics in this section describe how these tables are populated.

Contents

Phase 1 - Create Deposit Control

Phase 2 - Create Tender Control

Phase 3 - Create Payment Events, Tenders, Payments and Payment Segments

Phase 1 - Create Deposit Control

The following points describe, at a high level, the first phase of the payment upload process:

- PUPL checks that the record counts and money totals of tender control stagings add up to the expected amount on deposit control staging. If not,
 - PUPL sets the status of the deposit control staging to be **Error**. None of the tender controls within the deposit control will be processed until everything adds up. You can fix these on the [Deposit Control Staging](#) page.
 - When PUPL runs next, it will recheck the totals on deposit control stagings that are in **Error** or **Pending**
- If the record and dollar amounts are clean,
 - PUPL creates the corresponding deposit control
 - PUPL sets the status of the deposit control staging to be **In Progress**

Phase 2 - Create Tender Control

The following points describe, at a high level, the second phase of the payment upload process:

- PUPL checks that record counts and money totals of payment tender staging(s) adds up to expected amount on tender control staging. If not,
 - PUPL sets the status of the tender control staging to be **Error**. None of the tender controls within the deposit control will be processed until everything adds up for ALL tender controls. You can fix these on the [Tender Control Staging](#) page.

Note that the Deposit Control Staging record is **not** updated – its status is unchanged. Neither is the Pay Tender Staging record updated – its status also remains unchanged. Only the Tender Control Staging record is updated to be in **Error**.

- When PUPL runs next, it will recheck the totals of tender control stagings that are in **Error** or **Pending**.
- If the record and dollar amounts are clean,
 - PUPL creates the corresponding tender control.
 - PUPL sets the status of the tender control staging to be **In Progress**.

Phase 3 - Create Payment Events, Tenders, Payments and Payment Segments

At this point, all deposit control stagings and tender control stagings are in the state of **In Progress**. Next, PUPL starts the upload of payment tender staging and payment advice staging. The following points describe, at a high level, this phase of the payment upload process:

- If the payment tender staging record has a future accounting date, the processing for the record is skipped. This prevents uploaded payments from being created and subsequently frozen until their accounting date is reached. (Some external sources may provide advance notification of payments to be made in the future.) A skipped staging record remains in the **Pending** state until its accounting date is reached.
- PUPL checks money totals of payment advices (if any) adds up to expected amount on payment tender staging.
 - If not, PUPL sets the payment tender staging's status to **Error**.
 - Any errors are written to the [Payment Upload Exception](#) table. You can fix these errors on the [Payment Upload Staging](#) page and change the record's status back to **Pending**.
 - When PUPL runs next, it will recheck the totals of the payment tender staging
- If payment tender staging record is clean:
 - PUPL creates a corresponding payment event, tender, and payment.
 - If the account on payment tender staging is wrong, the account on the corresponding tender will be the tender source's suspense SA's account. Refer to [Setting Up Tender Sources](#) for more information. Refer to [How To Transfer A Payment From One Account To Another](#) for how to transfer to payment to the correct account.
 - If the account on payment advice is wrong, the account on the corresponding payment will be the account on the payment tender.
 - PUPL distributes the payment(s) amongst the account's service agreements, and payment segments are created. Note, the payment could be in error if there are no SA's for the account (as well as other reasons). Payments in error are written to the [Payment Exceptions](#) table.

- PUPL changes the payment tender staging's status to **Complete**.
- If all payment tender stagings are **Complete**:
 - PUPL changes the tender control staging's status to **Complete**.
 - PUPL changes the deposit control staging's status to **Complete**.
- If there are payment tender staging that are not **Complete**
 - The status of the tender control staging will still be **In progress**.
 - The status of the deposit control staging will still be **In progress**.
- PUPL will attempt to upload the offending payment tender staging records when it next runs.

PYUP-PRG - Purge Payment Upload Objects

Completed payment upload staging objects should be periodically purged from the system by executing the [PYUP-PRG](#) background process. This background process allows you to purge all **Completed** payment upload staging objects older than a given number of days.

We want to stress that there is no system constraint as to the number of **Completed** payment upload objects that may exist. You can retain these objects for as long as you desire. However we recommend that you periodically purge **Completed** payment upload objects as they exist only to satisfy auditing and reporting needs.

Maintaining Deposit Control Staging

The Deposit Control Staging page has three purposes:

- You can view historical deposit and tender control staging records associated with uploaded payments.
- You can correct deposit and tender control records that are in error.
- You can add deposit and tender control records to be uploaded by the payment upload background process.

The topics in this section describe this page.

Contents

[Deposit Control Staging - Main](#)

[Deposit Control Staging - Tender Control Staging](#)

Deposit Control Staging - Main

This page shows the details of a deposit control staging record.

Refer to [Populating The Payment Upload Staging Records](#) for more information about this record.

Open this page using **Financial, Deposit Control Staging**.

Description of Page

External Source ID corresponds with an external source ID on one of the your tender sources. This should be the unique ID of the source of the interfaced payments. Refer to [Setting Up Tender Sources](#) for more information.

External Transmit ID is the unique identifier of the transmission of payments from the external source. This must be a unique value for each transmission from the source.

Status shows the state of the deposit control staging records. Potential values are: *Incomplete, Pending, In Progress, Partial Load, Complete, Error*.

Deposit Control ID is the system-assigned, unique identifier of the related deposit control. This value is populated after the system creates a deposit control for the upload staging record.

Transmission Date/Time are when the information was interfaced into the system.

Total Tender Controls must equal the number of tender control staging records associated with this deposit control staging.

Total Tender Control Amount must equal the sum of the payment amounts on the tender control staging records associated with this deposit control staging. The **Currency Code** related to the amount is adjacent.

Deposit Control Staging - Tender Control Staging

This page shows the details of a tender control staging record.

Refer to [Populating The Payment Upload Staging Records](#) for more information about this record.

Open this page using **Financial, Deposit Control Staging, Tender Control Staging**.

Description of Page

External Source ID is the external source ID on the parent deposit control staging record.

External Transmit ID is the external transmission ID on the parent deposit control staging record.

The grid that follows contains a row for every tender control staging record linked to the deposit control staging record. The following information is displayed.

External Batch ID	This is the unique identifier of the batch of payments in respect of the external transmission ID.
Status	This is the state of the tender control staging records. Potential values are: <i>Pending, In Progress, Complete, Error</i> .
Tender Control ID	This is the system-assigned, unique identifier of the related tender control. This value is populated after the system creates a tender control for the upload staging record.
Total Tenders Amount	This is the sum of the payment amounts on the payment staging records associated with this tender control staging.
Total Number Of Tenders	This is the number payment tender staging records associated with this tender control staging.

Payment Upload Staging

The Payment Upload Staging page has three purposes:

- You can view historical payment tender and payment advice staging records associated with uploaded payments.
- You can correct payment tender records that are in error.
- You can add new payment tender and payment advice staging records to be uploaded by the payment upload process.

The topics in this section describe this page.

Contents

[Payment Upload Staging - Tender Detail](#)

[Payment Upload Staging - Payment Advice](#)

Payment Upload Staging - Tender Detail

This page shows the details of a payment tender staging record.

Refer to [Populating The Payment Upload Staging Records](#) for more information about this record.

Open this page using **Financial, Payment Upload Staging, Tender Detail**.

Description of Page

External Source ID this is the external source ID on the parent tender control staging record.

External Transmission ID is the external transmission ID on the parent tender control staging record.

External Batch ID is the external batch ID on the parent tender control staging record.

Ext. Reference ID is the external source's unique identifier of the payment tender.

Customer ID is the account ID or old account number of the customer tendering the payment. If the system cannot find an account ID or old account number that matches this value, the account ID of the tender source's suspense SA will be used on the corresponding tender and payment.

The **Tender Amount** is the amount tendered (i.e., the payment amount).

Tender Type defines the type of tender. Refer to [Setting Up Tender Types](#) for more information.

Authorizing Tenders. The **Tender Type** dropdown list does not include tender types that require authorization (i.e. credit card payments).

For information about tender types requiring authorization, refer to Credit Card Payments.

MICR ID is the MICR ID associated with the tender.

Check Number is the check number on the payment.

Name is the customer's name (as it appeared on the uploaded tender).

Accounting Date is the date that should be used for accounting purposes.

Pay Tender Staging Status shows the state of the tender control staging records. Potential values are: *Pending, Complete, Error*.

Payment Event ID is the system-assigned, unique identifier of the related payment event. This value is populated after the system creates a payment event for the upload staging record.

Payment Upload Staging - Payment Advice

If a tender was distributed to customers other than that defined on the Tender Detail page, the customers that the tender was distributed to are defined on this page. This page will not contain information if the tender is distributed to the tender detail's customer.

Refer to [Populating The Payment Upload Staging Records](#) for more information about this record.

Open this page using **Financial, Payment Upload Staging, Payment Advice**.

Description of Page

External Source ID this is the external source ID on the parent tender control staging record.

External Transmission ID is the external transmission ID on the parent tender control staging record.

External Batch ID is the external batch ID on the parent tender control staging record.

External Reference ID is the external reference ID of the payment tender.

The grid that follows is only populated if the tender is distributed to customer(s) other than the tendering customer. The following information is displayed.

Customer ID	This is the account ID or the old account number of the customer to which the payment should be distributed.
Customer Info	If the Customer ID is an account ID that exists in the system, the name of the primary person and the customer class of the account are displayed here.
Payment Amount	This is the amount of the tender to be distributed to the customer.
Match Type and Match Value	These fields are only used if the distribution of the payment should be restricted in some way (i.e., the standard payment distribution algorithm should not be used). Match Type indicates how the payment should be distributed (e.g., only distribute to a specific service agreement), Match Value indicates the ID of the restriction (e.g., the service agreement ID). Valid values of Match Type are <i>Service Agreement</i> .

Payment Upload Exception

If errors are detected during the payment upload process, a record is written to the payment upload exception table with a message indicating the nature of the severe error.

To view the messages associated with the exception records, schedule the [TD-PYUPL](#) background process. This process generates a To Do entry for every record in the payment upload exception table.

You can fix this error using the [Payment Upload Staging](#) page and change the status of the record from **Error** to **Pending**. When the payment upload process next runs, it attempts to upload this record again.

Interfacing Payments Using Distribution Rules

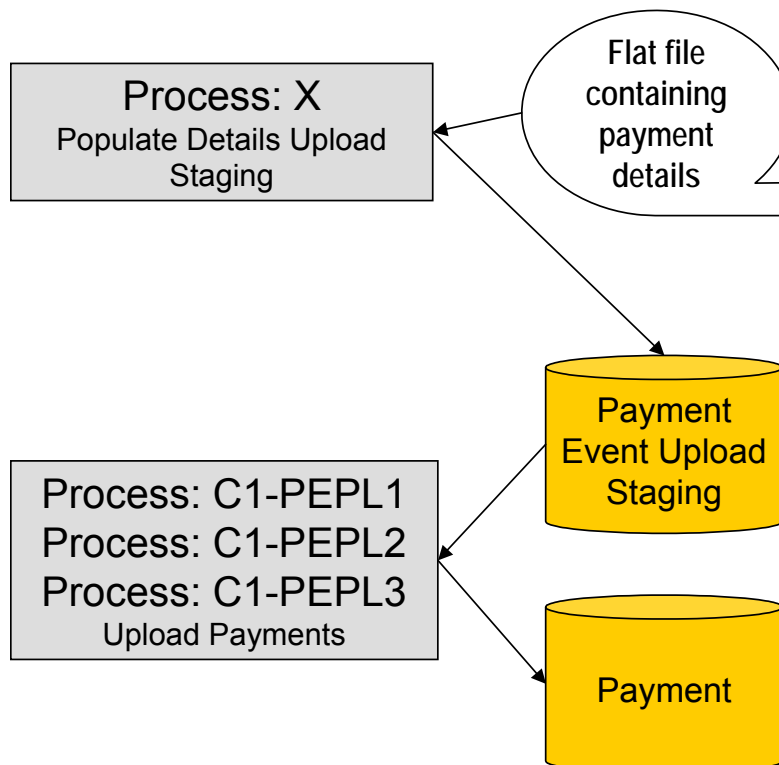
The topics in this section describe how the payment interface using distribution rules works.

Contents

[Populating The Payment Event Upload Staging Records](#)
[Payment Event Upload Staging](#)

Populating The Payment Event Upload Staging Records

The following diagram illustrates the processes involved in the uploading of payment event distribution details into the system.



The topics in this section describe each background process referenced above.

Contents

[Process X - Populate Payment Event Upload Staging](#)

[Data Setup Examples of Payment Distribution Details](#)
[The Lifecycle of a Payment Event Upload Staging Record](#)
[To Do Entries Instead of Exception Records](#)
[Process C1-PEPL1 - Upload Payments \(Step 1\)](#)
[Process C1-PEPL2 - Upload Payments \(Step 2\)](#)
[Process C1-PEPL3 - Upload Payments \(Step 3\)](#)

Process X - Populate Payment Event Upload Staging

Process X refers to the mechanism used by your organization to populate the payment event upload staging table.

Payment Event Upload Staging Table

You must create a payment event upload staging record for each payment distribution detail to be uploaded into the system. The name of this table is [CI PEVT DTL ST](#). The following table describes each column on this table.

Column Name	Length	Req'd	Data Type	Comments
EXT_SOURCE_ID	30	Y	A/N	This must correspond with an external source ID on one of the defined tender sources. Refer to Setting Up Tender Sources for more information.
EXT_TRANSMIT_ID	30	Y	A/N	This is the unique identifier of the transmission from the external source. This must be a unique value for each transmission from the source.
PEVT_DTL_SEQ	12	Y	N	Unique identifier of the detail record within the transmission. The C1-PEPL1 process uses this field to organize the parallel threads.
PEVT_STG_ST_FLG	4	Y	A/N	This must be set to 10 (10 is the lookup value that corresponds with the Incomplete state)
DST_RULE_CD	12	Y	A/N	This must be a valid distribution rule . The distribution rule contains the set of algorithms designed to process the staging detail.
DST_RULE_VALUE	254	Y	A/N	This must be a valid value for the characteristic type defined on the distribution rule.
CURRENCY_CD	3	Y	A/N	This must be a valid currency code (this would be USD for United States dollars).
TENDER_AMT	13.2	Y	N	The amount tendered (i.e., the payment amount).
ACCOUNTING_DT	10	Y	Date	This is the payment date that should be used for accounting purposes. This should correspond with an open accounting period.
TENDER_TYPE_CD	4	Y	A/N	This must be a valid tender type. Refer to Setting Up Tender Types for more information.
CHECK_NBR	10	N	A/N	This is the check number on the payment.
MICR_ID	30	N	A/N	This is the MICR ID associated with the payment tender.
CUST_ID	15	N	A/N	This field may be used to record customer information.

NAME1	30	N	A/N	This field may be used to capture additional payment tender information.
EXT_REFERENCE_ID	30	N	A/N	This field may be used to capture external information associated with the payment tender.
MATCH_TYPE_CD	8	N	A/N	See the description of the MATCH_VALUE field below. Refer to Payments And Match Events for more information about the significance of this field.
MATCH_VALUE	30	N	A/N	MATCH_VALUE and MATCH_TYPE_CD are used in conjunction to indicate that the distribution of the payment should be restricted in some way (i.e., the standard payment distribution algorithm should not be used). MATCH_TYPE_CD indicates how the payment should be distributed (e.g., only distribute to a specific service agreement), MATCH_VALUE contains the ID of the restriction (e.g., the SA ID). If MATCH_TYPE_CD is specified, it must reference a valid Match Type .
TNDR_CTL_ID	10	N	A/N	Leave this column blank. It will be assigned by the system when it creates a tender control record.
ACCT_ID	10	N	A/N	This is the tender account. If left blank, the <i>C1-PEPL1</i> process will populate this field by calling the <i>Determine Tender Account</i> algorithm defined on the distribution rule . Note that this Account ID is not necessarily unique as multiple staging details can reference the same tender account.
PAY_EVENT_ID	12	N	A/N	Leave this column blank. It will be assigned by the system when it creates a payment event record.
PEVT_PROCESS_ID	10	N	A/N	If left blank, the <i>C1-PEPL1</i> process will set this field equal to the Tender Account ID. This field is used for grouping staging records and for organizing parallel threads (in the <i>C1-PEPL2</i> process). Therefore, it is strongly encouraged that it bears a relationship to the tender account ID.
APAY_SRC_CD	12	N	A/N	This must be a valid auto-pay source code.
EXT_ACCT_ID	50	N	A/N	This is the customer's account number at the financial institution
EXPIRE_DT	10	N	Date	This field is only needed if the Tender Type indicates that an expiration date is necessary (e.g., for a credit card payment)
ENTITY_NAME	64	N	A/N	This is the customer's name in the financial institution's system

Data Setup Examples of Payment Distribution Details

Typically, each staging record will represent a single payment event – with a corresponding payment tender and payment. However, it is possible to specify complex relationships within a set of staging records. For example, it will be straightforward to define a set of staging records that represent a single tender but multiple payments (to model the single payment tender of a welfare agency which covers payments for multiple accounts). Similarly, it is equally possible to define a set of staging records that represent a single payment but multiple tenders (although not a common requirement).

Each staging record can represent:

- Zero or one new payment events. Typically, each detail staging record will represent a single payment event. However, it is possible to define multiple records for a single payment event. All details for a single payment event are identified by a common value on the staging record: **Pay Event Process ID**.

Pay Event Process ID. This field is used for grouping staging records **and** for organizing parallel threads (in the **C1-PEPL2** process). Therefore, it is **strongly encouraged** that it bears a relationship to the tender account ID.

- A partial or one new payment tenders. Typically, each detail staging record will represent a single payment tender. However, it is possible to define multiple staging records for a single pay tender. All details for a single tender are identified by a common payment event ID as well as common tender information (tender type, tender account ID, check number, external reference, etc.).
- A partial or many new payments. Typically, each detail staging record will represent a set of one or many new payments. However, it is possible to define multiple staging records for a single payment. All details for a single payment are identified by a common payment event ID as well as common distribution rule and characteristic value information.

The sections below provide examples of a few of these complex payment event configurations. Note that these examples assume the same distribution rule is referenced in all staging records.

Contents

[Staging Entry Example 1: One Payment Event, Two Tenders, Two Payments](#)

[Staging Entry Example 2: One Payment Event, One Tender, Two Payments](#)

[Staging Entry Example 3: One Payment Event, Two Tenders, One Payment](#)

Staging Entry Example 1: One Payment Event, Two Tenders, Two Payments

Seq.	Pay Event Process ID	Tender Account ID	Tender Ref.	Date	Rule Value	Amount	Tender Type	Check No
1	1234567890	1234567890	112A	1/1/2006	113-54-8978	30.00	Check	101
2	1234567890	1234567890	112B	1/1/2006	575-40-3030	40.00	Check	102

Staging Entry Example 2: One Payment Event, One Tender, Two Payments

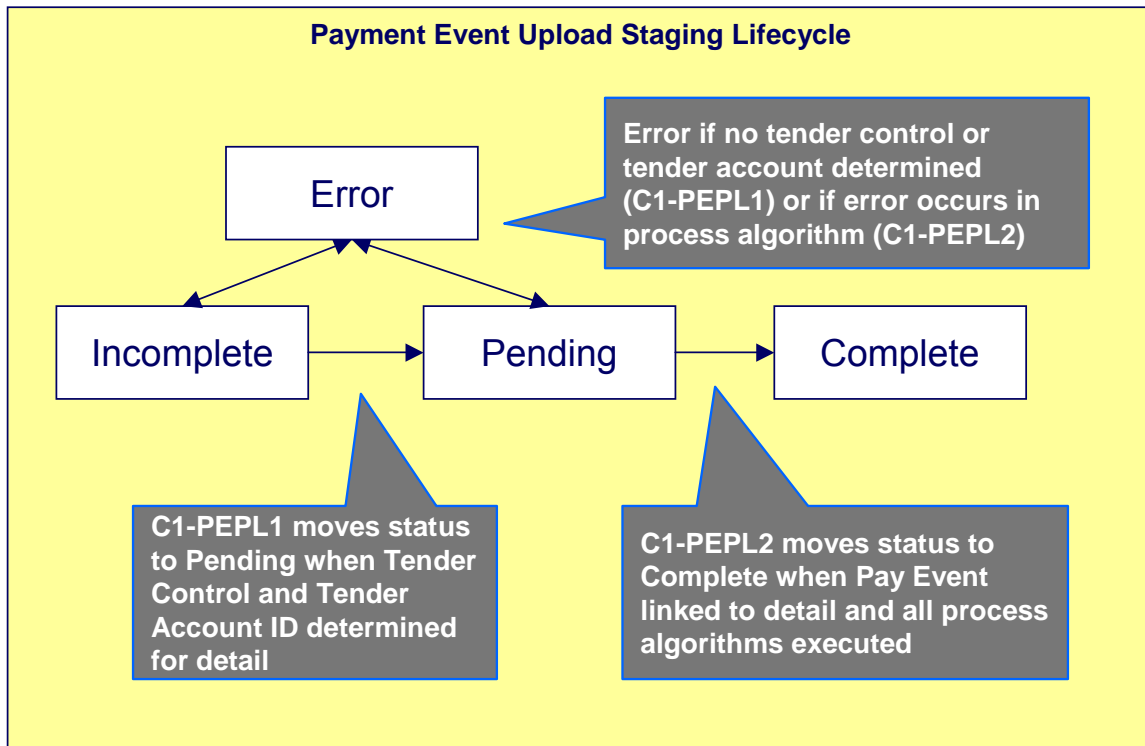
Seq.	Pay Event Process ID	Tender Account ID	Tender Ref.	Date	Rule Value	Amount	Tender Type	Check No
1	1234567890	1234567890	112A	1/1/2006	113-54-8978	30.00	Check	101
2	1234567890	1234567890	112A	1/1/2006	575-40-3030	40.00	Check	101

Staging Entry Example 3: One Payment Event, Two Tenders, One Payment

Seq.	Pay Event Process ID	Tender Account ID	Tender Ref.	Date	Rule Value	Amount	Tender Type	Check No
1	1234567890	1234567890	112A	1/1/2006	575-40-3030	30.00	Check	101
2	1234567890	7878787870	888Q	1/1/2006	575-40-3030	40.00	Check	9872

The Lifecycle of a Payment Event Upload Staging Record

The following diagram shows the possible lifecycle of a payment event upload staging record.



- **Incomplete.** A payment event staging record is initially created in **incomplete** state. The **C1-PEPL1** process sets it to **pending** once it links it to a tender control and determines its tender account.
- **Pending.** The **C1-PEPL2** process sets a **pending** record to **complete** once all processing logic is executed and a payment event is linked to it.
- **Complete.** When processing of the staging record is complete the record is in the **complete** state. This is a final state.
- **Error.** A payment event staging record may be set to **Error** from **Incomplete** or **Pending** states by the **C1-PEPL1** and **C1-PEPL2** processes respectively.

Refer to [To Do Entries Instead of Exceptions](#) for more information on how To Do entries are used to capture processing errors.

To Do Entries Instead of Exception Records

Instead of creating an exception record for staging records in the **Error** state the **C1-PEPL1** and **C1-PEPL2** background processes create To Do entries and link them to the offending staging records.

Each process determines the **To Do type** to use for reporting errors by looking for the To Do Type defined with this process as its creation process.

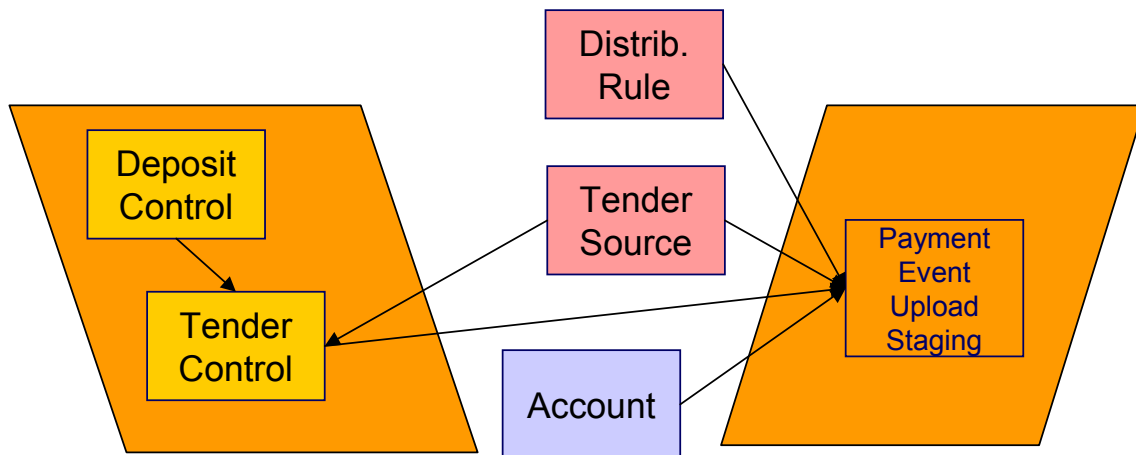
Process C1-PEPL1 - Upload Payments (Step 1)

The batch process **C1-PEPL1** refers to the first of three background processes that load the contents of the payment event upload staging records into the various payment tables.

The responsibility of this process is to transition the status of the staging records from **Incomplete** to **Pending**. The status of a staging record must remain **Incomplete** until:

- It is linked to a tender control. This process creates new deposit and tender control records, and then updates the payment event upload staging records with the corresponding **Tender Control ID**.
- The **Tender Account ID** field is populated. The **Determine Tender Account** algorithm defined on the **distribution rule** is called and the returned account ID is posted on the staging record.
- The **Pay Event Process ID** field is populated. If left blank this field is set equal to the **Tender Account ID**.

The following diagram and the sections below describe at a high level the processing phases of the **C1-PEPL1** background process.



Contents

Phase 1 - Create Tender Control

Phase 2 - Determine Tender Account

Phase 1 - Create Tender Control

Note. This step cannot be bounded by thread range but must execute across the entire population of staging details. Therefore this step in the process is designed so that all parallel threads attempt to execute it at the same time but only one thread succeeds to avoid creating duplicate tender controls.

This step creates all tender controls required by the upload records as follows:

- For each distinct tender source transmissions represented within the **incomplete** set of upload staging details a deposit control, a deposit tender and a tender control are created in an **open** state. Note that a deposit tender record is only created if the tender amount is not zero.
- If an error occurs at this stage
 - A designated staging record for the transmission group (one is picked at random) is set to **Error** and a To Do entry is created and linked to it to capture the error message that applies for the whole transmission group. Other records in the group remain **incomplete**.
 - The process stops.

Group Error. This technique allows for an easier recovery from a setup error that may affect a large volume of records in a single transmission. Capturing the error only on a single designated record requires only this record to be set back to **Incomplete** once the setup issue is corrected. It is important to note that the transmission group is not processed if at least one of the records in the group is in **Error** status.

- Once a transmission group of records is fully processed, a To Do cleanup processing takes place to complete To Do entries previously raised for its designated staging record.

Refer to [To Do Entries Instead of Exceptions](#) for more information on how To Do entries are used to capture processing errors.

Phase 2 - Determine Tender Account

After all tender controls have been created the second step attempts to transition all **incomplete** records to the **pending** state.

Each **incomplete** staging record is processed as follows:

- If not yet associated with a tender control ID, the process looks for an **Open** tender control that matches the batch code, batch number and external source ID and links it to the staging record. An error is raised if a matching tender control is not found.
- Execute the **Determine Tender Account** algorithm defined on the distribution rule and populate the record with the returned account ID. Note that the algorithm is called even when the tender account is populated to provide for a potential override of the initial value when necessary. An error is raised if a tender account may not be determined.
- If not already populated, the **Pay Event Process ID** is set equal to the Tender Account ID. The **C1-PEPL2** process uses this field to organize the parallel threads and to group multiple staging details into a single payment event.
- The staging record is moved to **pending** state.
- If any errors occur set the record to **Error** and create a To Do entry for the error message.
- If no error occurred, a To Do cleanup processing takes place to complete To Do entries previously raised for the staging record.

Refer to [To Do Entries Instead of Exceptions](#) for more information on how To Do entries are used to capture processing errors.

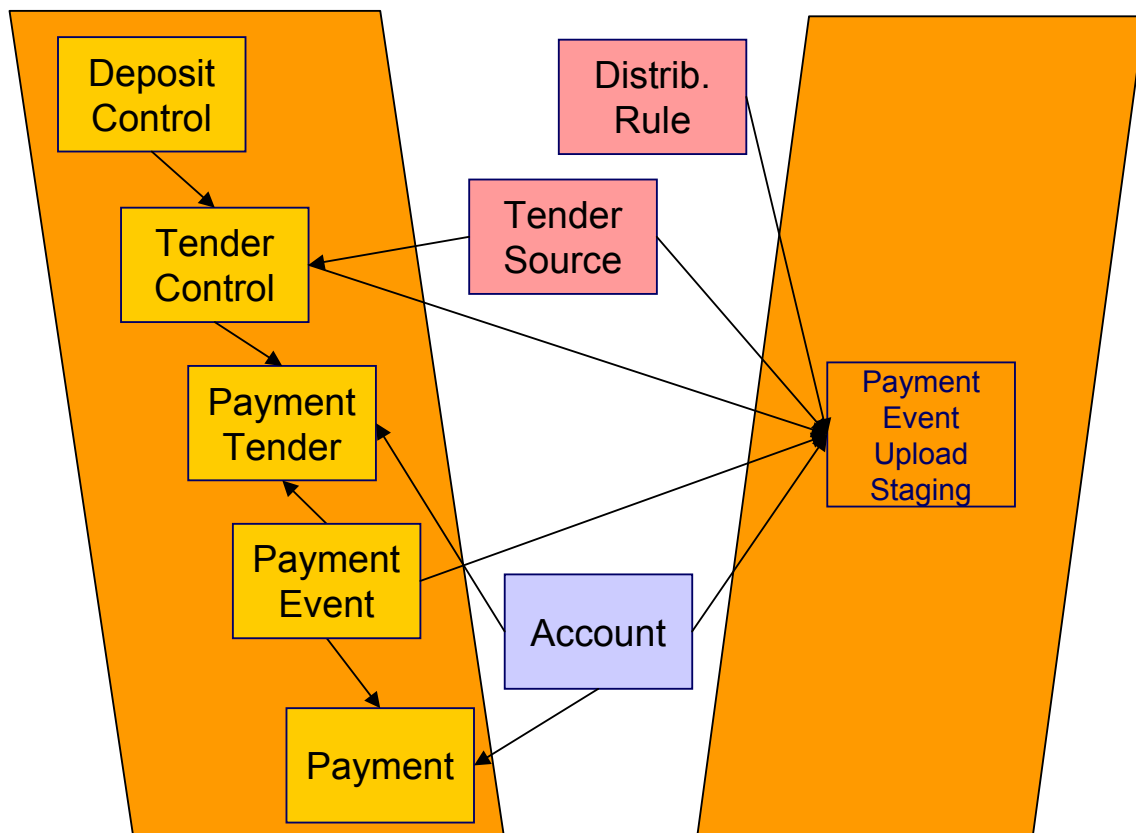
Note. This step is designed to support execution in parallel threads based upon the sequence number portion of the staging table prime key.

Process C1-PEPL2 - Upload Payments (Step 2)

The batch process **C1-PEPL2** refers to the second of three background processes that load the contents of the payment event upload staging records into the various payment tables.

The responsibility of the **C1-PEPL2** process is to create payment events, payment tenders and payments and transition the corresponding staging records from **Pending** to **Complete**.

The following diagram and the section below describe at a high level the processing phases of the **C1-PEPL2** background process.



Each distinct group of **pending** staging records associated with the same external source ID, external transmit ID, accounting date, and pay event process ID is processed as follows:

- A payment event is created for the group and stamped on each of its records.
- A payment tender is created for each distinct set of staging records having the same tender account, tender type and other tender information fields, except for the tender amount.

- The **Create Payment** algorithm is then called for each distinct group of staging records having the same tender account, [distribution rule](#) and rule value providing it with the total amount for the group.
- The staging record is moved to **pending** state.
- If any error occurs a designated staging record for the payment event group (one is picked at random) is set to **Error** and a To Do entry is created and linked to it to capture the error message that applies for the whole group. Other records in the group remain **incomplete**.

Group Error. This technique allows for an easier recovery from an error that affects all staging records for a single payment event. Capturing the error only on a single designated record requires only this record to be set back to **pending** once the issue is corrected. It is important to note that the whole set of records is not processed if at least one of the records in the group is in **Error** status.

- If no error occurred, a To Do cleanup processing takes place to complete To Do entries previously raised for the designated staging record.

Refer to [To Do Entries Instead of Exceptions](#) for more information on how To Do entries are used to capture processing errors.

Note. This process is designed to support execution in parallel threads based upon the payment event process ID field.

Process C1-PEPL3 - Upload Payments (Step 3)

The batch process **C1-PEPL3** refers to the last of three background processes that load the contents of the payment event upload staging records into the various payment tables.

The responsibility of the **C1-PEPL3** process is to update the status of the related deposit and tender controls from **open** to **balanced**.

Each distinct tender control for which all associated staging records are in **complete** status is processed as follows:

- The tender control is set to **balanced**.
- The deposit control is set to **balanced**.

Note. This process is designed to support execution in parallel threads based upon the Tender Control ID field.

Payment Event Upload Staging

The Payment Event Upload Staging page has three purposes:

- You can view historical payment event upload staging records associated with uploaded payments.
- You can correct payment event upload staging records that are in **error**.

- You can add new payment event upload staging records to be uploaded by the payment event upload processes.

The topics in this section describe this page.

Payment Event Upload Staging - Main

This page shows the details of a payment event upload staging record.

Refer to [Populating The Payment Event Upload Staging Records](#) for more information about this record.

Open this page using **Financial, Payment Upload Staging, Tender Detail**.

Description of Page

External Source ID corresponds with an external source ID on one of the your tender sources. This should be the unique ID of the source of the interfaced payments. Refer to [Setting Up Tender Sources](#) for more information.

External Transmission ID is the unique identifier of the transmission of payments from the external source. This must be a unique value for each transmission from the source.

Sequence is the identifier of the record within the transmission. The **C1-PEPL1** process uses this field to organize the parallel threads.

Accounting Date is the payment date that should be used for accounting purposes.

Distribution Rule is the rule by which the payment detail is to be processed. A [default distribution rule](#) is displayed if you have set one.

Rule Value is a value associated with the payment and expected by the distribution rule.

Match Type and **Match Value** are only used if the distribution of the payment should be restricted in some way (i.e., the standard payment distribution algorithm should not be used).

Match Type indicates how the payment should be distributed (e.g., only distribute to a specific service agreement), **Match Value** indicates the ID of the restriction (e.g., the service agreement ID).

Payor Account ID is the tender account. If not populated the **C1-PEPL1** process populates this field by calling the **Determine Tender Account** algorithm defined on the [distribution rule](#).

Tender Amount is the amount tendered (i.e., the payment amount).

Currency Code is the currency of the tendered amount. This should be the same as currency defined on the tender source.

Tender Type defines the form or remittance (e.g., cash, check, etc.). Note that the **Tender Type** defaults from the [installation record](#).

Authorizing Tenders. The **Tender Type** dropdown list does not include tender types that require authorization (i.e. credit card payments).

For information about tender types requiring authorization, refer to [Credit Card Payments](#).

Check Number is the check number on the payment.

MICR ID is the value of the magnetic ink character recognition (MICR) line on the payment.

External Reference ID may be used to record external information associated with the payment tender.

Customer ID may be used to record additional customer information.

Name may be used to record additional payment tender information.

Tender Control ID is the tender control associated with the payment. This field is should typically be left for the **C1-PEPL1** process to populate.

If the **Tender Type** is associated with an [automatic payment](#), the **Auto Pay** section displays. The system attempts to default automatic payment information from the [account's auto-pay](#) option if the tender type is the same as the tender type on the account's auto-pay source and if the auto pay option is effective on the payment date. If the system is unable to default information, you must specify the source of the funds and the customer's account number / credit card number at the financial institution.

- **Auto Pay Source Code** is the financial institution / credit card company that receives the automatic payment request.
- **External Account ID** is the customer's account number at the financial institution.
- **Expires On** is only needed if the **Tender Type** indicates that an expiration date is necessary (e.g., for a credit card payment).
- **Name** is the customer's name in the financial institution's system.

Pay Event Process ID is used to group multiple staging records into a single payment event. If not populated, the **C1-PEPL1** process sets this field equal to the **Payor Account ID**.

Pay Event Staging Status shows the state of the staging record. Refer to [The Lifecycle of a Payment Event Upload Staging](#) for a state transition overview.

Payment Event ID is the system-assigned, unique identifier of the related payment event. The **C1-PEPL2** process populates this field when it creates a payment event for the upload staging record. You can use this field to navigate to the payment event page.

If a staging record is in **Error** state then the error message associated with the [corresponding To Do entry](#) is displayed.

Adjustments

In this section, we describe when and how to adjust a service agreement's balance.

Contents

- [The Big Picture Of Adjustments](#)
- [Maintaining Adjustments](#)
- [How and When To Use An Adjustment](#)
- [Interfacing Adjustments From External Sources](#)

The Big Picture Of Adjustments

The topics in this section provide background information about a variety of adjustment issues.

We strongly recommend familiarizing yourself with the topics described in [The Financial Big Picture](#) to fully appreciate the place of an adjustment in the system's financial architecture. In particular, refer to [Setting Up Adjustment Types](#) and [Setting Up Adjustment Type Profiles](#).

Contents

- [Adjustments - Current Balance versus Payoff Balance](#)
- [Canceling Adjustments](#)
- [Transfer Adjustments](#)
- [Calculated Adjustments](#)
- [Adjustment Amount May Be Positive, Negative Or Zero](#)
- [Adjustment Type Controls Everything](#)
- [Unbilled Adjustments And Aged Debt](#)
- [An Adjustment May Affect More Than Just Customer Balances](#)

Adjustments - Current Balance versus Payoff Balance

Adjusting how much a customer owes involves changing a service agreement's payoff balance and/or current balance by creating an adjustment. In this section we describe these two balances.

Warning! If you do not understand the difference between payoff balance and current balance, refer to [Current Amount versus Payoff Amount](#).

Contents

- [When Current Balance Equals Payoff Balance](#)
- [When Current Balance Differs From Payoff Balance](#)
- [Adjustment Type And Balances](#)

When Current Balance Equals Payoff Balance

For most service agreements, payoff balance and current balance are always the same (or in colloquial speech - the amount the customer thinks they owe equals what they really owe). In this situation, an adjustment is easy: both payoff balance and current balance are adjusted by the same value.

Let's run through a typical example. The values in the payoff balance and current balance columns reflect the amount due after the financial transaction has been applied (i.e., the running balance):

Date	Financial Transaction	Payoff Balance	Current Balance
1-Jan-99	Bill: \$125	125	125
15-Jan-99	Payment: \$150	-25	-25
2-Feb-99	Bill: \$175	150	150
14-Feb-99	Payment: \$150	0	0
3-Mar-99	Bill: \$200	200	200
15-Mar-99	Payment: \$150	50	50
2-Apr-99	Bill: \$225	275	275
27-Apr-99	Adj: Late Payment Charge \$10	285	285

As you can see, payoff balance and current balance are always in sync.

When Current Balance Differs From Payoff Balance

For some service agreements, payoff balance and current balance differ (or in colloquial speech - the amount the customer thinks they owe differs from what they would owe if they wanted to payoff their account).

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

Date	Financial Transaction	Payoff Balance	Current Balance
1-Jan-99	Bill: \$125, Budget \$150	125	150
15-Jan-99	Payment: \$150	-25	0
2-Feb-99	Bill: \$175, Budget \$150	150	150
14-Feb-99	Payment: \$150	0	0
3-Mar-99	Bill: \$200, Budget \$150	200	150
15-Mar-99	Payment: \$150	50	0
2-Apr-99	Adj: Mar 3, 97 bill incorrect due to meter exchange discrepancies; manual correction of -\$33.	17	0

Notice the adjustment of -\$33 affects only payoff amount. This is because the customer is on a budget. The adjustment will not affect their monthly payment and therefore it doesn't affect how much they think they owe.

The following table describes the various types of service agreements where these balances may differ.

Type Of Service Agreement	What Payoff Balance Holds	What Current Balance Holds
---------------------------	---------------------------	----------------------------

Deposit	Payoff balance holds the amount of deposit you are holding on behalf of the customer's debt. This amount is showed as a credit (with a negative sign) because you truly owe the customer this money.	Current balance holds the amount the customer owes for a deposit.
Charitable contribution	N/A - charitable contribution service agreements never have a payoff balance.	Current balance holds the amount the customer owes for a charitable contribution.
Payment arrangements	Payoff balance holds the total amount of delinquent debt that remains unpaid. When a payment arrangement is started, you transfer the payoff balance from the utility service agreement to the payment arrangement service agreement. Each billing period, the customer pays down this balance.	Current balance holds the amount the customer owes in respect of the billed payment arrangement amounts.
Utility agreements on a budget plan	Payoff balance holds the actual amount owed (actual bills minus payments).	Current balance holds the amount the customer owes in respect of the billed budget amount.
Loans	Payoff balance holds the payoff amount for the loan, which is the current balance (the billed loan amount) plus the principal balance (the unbilled loan amount) and any accrued interest charges.	Current balance holds the amount the customer owes in respect of the billed amount (the periodic payment amount).

Adjustment Type And Balances

When you create an adjustment, you must define its adjustment type. The adjustment type controls how payoff balance and current balance are affected by the adjustment amount.

You can only pick adjustment types that make sense for the service agreement. For example, if your organization doesn't allow budget billing, all adjustments for a utility service agreement will affect payoff balance and current balance the same (you can't get the balances out-of-sync).

The various types of adjustments that may be linked to a service agreement are controlled by the adjustment profiles defined on the service agreement's SA type.

Refer to [Adjustment Type Controls Everything](#) for more information.

Canceling Adjustments

You may cancel any frozen adjustment. Canceling the adjustment creates another financial transaction that reverses the financial effects of the original adjustment. The impact of the cancellation appears on the customer's next bill. You may view both the original financial transaction and its cancellation on the Adjustment page.

Transfer Adjustments

A convenient mechanism exists to transfer moneys between two service agreements. The net effect of such a request is the creation of two adjustments, each of which is linked to the other. Both adjustments are created together, frozen together, and posted to the GL together. This is useful because you can't create one side of the transfer and without the other. Refer to [How To Create A Transfer Adjustment](#) for more information.

Inter or Intra Account. It's important to be aware that the transfer can be inter or intra account (i.e., the accounts on the two adjustments may be different).

Calculated Adjustments

An adjustment can calculate the adjustment amount using an algorithm. A calculated adjustment can be used to calculate:

- Sales or other taxes for a base amount
- Any charge amount based on a user-supplied quantity, such as feet, miles, hours, etc.
- Others.... Any calculation that can be made by a rate can be applied to an adjustment.

When you create a calculated adjustment, you may be asked to supply a base amount. The user may supply the base amount or the amount could be [defaulted](#) from the adjustment type and possibly overridden by the user prior to calculating the adjustment amount.

When the adjustment is generated, the calculation algorithm may use the base amount and calculation date. The base package algorithm calls the rate application and returns the calculated amount with calculation lines and GL distribution codes.

An adjustment's adjustment type controls whether the adjustment is calculated and the algorithm that performs the calculation.

For more information, refer to [Adjustment Type Controls Everything](#) and [Setting Up Calculated Adjustment Types](#).

Adjustment Amount May Be Positive, Negative Or Zero

An adjustment's amount may be positive, negative or zero:

- A positive amount causes the customer's balance(s) to increase.
- A negative amount causes the customer's balance(s) to decrease.
- A zero amount will not affect the customer's balance(s). A zero amount is odd, but necessary when you need to use an adjustment to correct the GL distribution. Refer to [How To Use An Adjustment To Change Amounts Booked In Your GL](#) for more information.

Adjustment Type Controls Everything

When you create an adjustment, you must define its adjustment type. The topics in this section describe how adjustment type controls the behavior of an adjustment.

Contents

- [Controls Which Balance\(s\) Are Affected](#)
- [Defines The GL Account Affected By The Adjustment](#)
- [Defaults The Adjustment Amount](#)
- [Calculates The Adjustment Amount](#)
- [Controls The Interface To A/P & 1099 Reporting](#)
- [Controls Information Printed On The Bill](#)

[Controls If The Adjustment Can Be Frozen Prior To Bill Completion](#)
[Controls If The Adjustment Requires Approval](#)

Controls Which Balance(s) Are Affected

The adjustment type's financial transaction (FT) algorithm controls how payoff balance and current balance are affected by the adjustment amount.

Defines The GL Account Affected By The Adjustment

Most adjustments affect the general ledger (GL) in some way. The following points describe the source of these GL accounts.

- For normal adjustments (i.e., non-calculated adjustments that affect a single service agreement), there is a single accounting entry generated:
 - One side of the accounting entry is taken from the distribution code on the SA type of the service agreement affected by the adjustment. For example, if you are adjusting the payoff balance on a normal service agreement, the A/R account is constructed from the distribution code on the service agreement's SA type.
 - The other side of the accounting entry is taken from the distribution code on the adjustment's adjustment type. For example, if a late payment charge is created, the "late payment charge" adjustment type references this fee's revenue account.
- For transfer adjustments (i.e., adjustments used to transfer moneys between two service agreements), there are two accounting entries generated - one for the "from" side and one for the "to" side. Each adjustment carries its own set of balanced GL accounting details.
 - For each adjustment, one side of the entry is taken from the distribution code on the SA type of the service agreement affected by the adjustment (just like for normal adjustments).
 - The other sides of both accounting entries have the same GL account. This account should be the intermediate clearing GL account that is to be used for the transfer. The source of this clearing GL account is the distribution code on the adjustment type used to transfer the funds.
- For calculated adjustments (i.e., adjustments where rates may be applied to calculate the adjustment amount), there is a single accounting entry:
 - For each adjustment, one side of the entry is taken from the distribution code on the SA type of the service agreement affected by the adjustment (just like for normal adjustments).
 - The other side of the entry is taken from the distribution codes that resulted from the rate calculation if the adjustment financial transaction algorithm is set up to use the calculation lines as the distribution code source. If the adjustment financial transaction algorithm is set up to use the adjustment type as the distribution code source, the other side of the accounting entry is taken from the distribution code on the adjustment type.

Not all adjustments affect the GL. As a general rule of thumb, only those adjustments that affect a service agreement's payoff balance affect the GL. So, if you adjust a budget customer's current balance, but don't adjust their payoff balance, the GL is not affected.

Defaults The Adjustment Amount

The adjustment type may default the adjustment amount in one of the following ways:

- The adjustment type may specify a default amount. This would be used for those adjustment types that have a standard charge for all customers that receive this adjustment, for example a non-sufficient funds charge.
- The adjustment type may specify a default adjustment amount algorithm. This would be used for those adjustment types that have a charge that varies based on other factors. For example, a non-sufficient funds charge may be based on a customer's credit rating.

When an amount is defaulted onto a new adjustment it may be overridden by a user.

Calculates The Adjustment Amount

Some adjustment types [calculate the adjustment](#) amount. If the adjustment type uses a calculated amount, a base amount is passed to the rate application, which then returns the calculated amount with calculation lines and GL distribution codes.

The adjustment type's generate adjustment algorithm controls which rate is applied to the base amount. A user supplied calculation date controls which version of the rate is used. The user may supply the base amount or it may be [defaulted](#) from the adjustment type and possibly overridden by the user prior to calculating the adjustment amount.

Controls The Interface To A/P & 1099 Reporting

If the adjustment type is associated with a payment (e.g. a refund or a loan disbursement) of money to a customer, the adjustment type indicates such with a reference to an A/P request type.

When an adjustment that references an A/P request type is **frozen**, an A/P download request record is created. This record is the interface request to ask your A/P system to cut a check. This interface record is marked with a batch process ID and run number.

- The batch process ID is the process responsible for creating the flat file that contains check request that is interfaced to your account's payable system. The batch process ID is defined on [Installation Options – Financial](#). The base package is supplied with a skeletal background process (referred to by the process ID of **APDL**) that must be populated with logic to format the records in the format compatible with your accounts payable system.
- The run number is the batch process ID's current run number.

Refer to Accounts Payable Check Request for more information.

If the resultant check needs to be reported for income tax purposes under a specific 1099 category, the category is also specified on the adjustment type. The 1099 category is in turn interfaced to the A/P system (the system does NOT manage 1099 reporting).

Controls Information Printed On The Bill

If the adjustment appears on the customer's next bill, the verbiage is specified on the adjustment type (and may NOT be overridden on the adjustment).

Controls If The Adjustment Can Be Frozen Prior To Bill Completion

Refer to [Preventing SA Balances And The GL From Being Impacted Until Bill Completion](#) for how adjustment type can control whether adjustments of a given type can be frozen prior to bill completion.

Controls If The Adjustment Requires Approval

If an adjustment's adjustment type references an approval profile, the system will not allow the initiating user to freeze the adjustment. Rather, the initiating user can submit the adjustment for approval. When an adjustment is submitted for approval, the system determines the necessary approval levels and notifies the first approver. The system will freeze the adjustment when last approver approves the adjustment (if the adjustment's adjustment type allows it to be frozen prior to the completion of the next bill).

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

Refer to [The Big Picture of Adjustment Approvals](#) for more information.

Unbilled Adjustments And Aged Debt

A **Frozen** adjustment waits in limbo until the customer's next bill is produced. This limbo period could be several weeks if the customer is billed infrequently. When the customer's next bill is completed, all limbo adjustments are linked to the bill. A brief description (as defined on the adjustment type) appears on the printed bill.

It is important to stress the following in respect of these limbo adjustments:

- If the adjustment decreases the amount of debt, the customer's aged debt is affected immediately regardless of whether the adjustment appears on a bill.
- If the adjustment increases the amount of debt, the amount the customer owes from an aged debt perspective may or may not be affected by the adjustment. There is a switch on the financial transaction created for the adjustment called New Charge that controls the arrears behavior. When the FT for this type of adjustment is created, the switch is set to on and the customer's aged debt will not reflect the adjustment amount until the adjustment is swept onto a bill. The moment the adjustment is swept onto the customer's bill, the debt starts aging. A user may turn the switch off to cause the debt to start aging sooner. For this case, the date on which the adjustment starts aging must be defined in the Arrears Date field.
- The amount a customer owes in total is immediately affected by the adjustment regardless of whether the adjustment appears on a bill. This means that amount of aged debt may not be in sync with the total amount owed. This seems odd but is useful from a credit and collections perspective. You see, you probably don't want to start aging an adjustment until the customer has actually seen it.

An Adjustment May Affect More Than Just Customer Balances

When an adjustment is frozen or cancelled, a customer's current and payoff balances are affected. However, several other objects may be affected when such events occur. Refer to [Obscure Things That Can Happen](#) for more information.

Maintaining Adjustments

An adjustment is used to change the amount of debt stored on a service agreement. The topics in this section describe the pages on which adjustments are maintained.

For more information about adjustments, refer to [The Big Picture Of Adjustments](#).

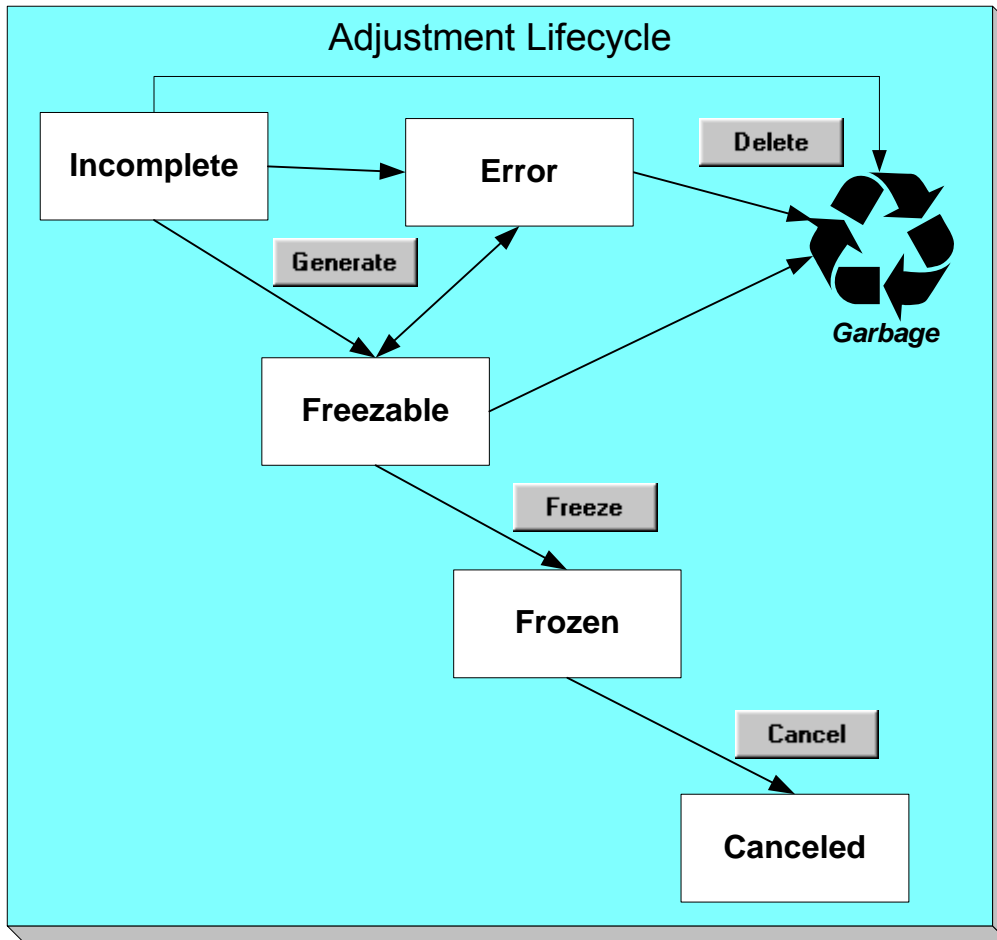
Contents

- [The Lifecycle Of An Adjustment](#)
- [Adjustments - Main Information](#)
- [Adjustments - Characteristics](#)
- [Adjustments - Transfer Adjustment](#)
- [Adjustments - A/P Request](#)
- [Adjustments - Approval](#)
- [Financial - Adjustment Calculation Line Characteristics](#)

The Lifecycle Of An Adjustment

The following diagram shows the possible lifecycle of an adjustment.

Warning! This diagram only makes sense in the context of the page used to maintain Adjustments. Refer to [Adjustments - Main Information](#) for the details.



Adjustments are initially created in the **Incomplete** state. Adjustments in this state don't have a financial transaction. This means, you can change the adjustment type and amount at will.

Click **Generate** to generate a financial transaction for the adjustment. The financial transaction contains the adjustment's effect on the general ledger and on the customer's payoff and current balances. If the adjustment is calculated, the algorithm on the adjustment type's Generate Adjustment event controls how the adjustment is calculated. If the adjustment is calculated, you must specify the calculation date, which is passed as a parameter to the calculation algorithm and is used for calculations that are effective dated (e.g., rate version or bill factor value). The financial transaction (FT) algorithm defined on the adjustment type Adj. Financial Transaction event controls how the financial transaction is constructed. For calculated adjustments, the distribution code may be taken from the adjustment type or the calculation lines.

After generating the financial transaction, the adjustment becomes **Freezable**. While in this state, you may change the service agreement, adjustment type and amount at will. However, if you change this information, you will have to regenerate the financial transaction (by clicking the Generate button).

In the very rare situation when the system cannot generate the financial transaction because of inconsistent setup data, the adjustment is moved to the **Error** state. You may regenerate the financial transaction after correcting the source of the error. You may also delete such an adjustment.

Click the delete button to physically remove an **Incomplete**, **Error** or **Freezable** adjustment from the database.

Freeze may not be enabled. Please be aware that you can prevent specific types of adjustments from being frozen until the account's next bill is **completed**. Refer to [Preventing SA Balances And The GL From Being Impacted Until Bill Completion](#) for how to enable this feature on your adjustment types.

Click **Freeze** to freeze the adjustment and its financial transaction. After doing this, the adjustment's state becomes **Frozen**. While in this state, you cannot change the adjustment's type or amount, but you may change:

- When the adjustment starts aging.
- Whether the adjustment appears on a customer's bill.
- The accounting date used to derive the general ledger accounting period(s) to which the financial transaction is booked.

Adjustments may require approval. If the adjustment type referenced on the adjustment has an approval profile, the Freeze button is replaced with a Submit for Approval button. If an adjustment is currently being approved, all action buttons are disabled because the adjustment must be either approved or rejected using the [Adjustments - Approval](#) page. Refer to [The Big Picture of Adjustment Approvals](#) for more information.

If you need to remove the financial effects of an adjustment, click **Cancel**. Canceling an adjustment causes the generation of another financial transaction. This new financial transaction reverses the financial impact of the original adjustment.

Adjustments - Main Information

The **Description of Page** section below describes the fields on this page. Refer to [How and When To Use An Adjustment](#) for instructions describing how to perform common maintenance functions.

The Main page contains core adjustment information. Open this page using **Financial, Adjustment**.

Warning! All adjustments reference a very important field that controls much validation and processing; this field is called Adjustment Type. Take special care when adding a new adjustment to specify the appropriate Adjustment Type as it affects how the adjustment appears on the customer's bill, how the adjustment is reflected in your general ledger, and much more. After the adjustment is frozen, you may not change its Adjustment Type.

Description of Page

Adjustment Info contains a concatenation of the adjustment amount, adjustment type and status.

Formatting may be performed by a plug-in. The contents of **Adjustment Info** may be formatted by a plug-in algorithm on the [Adjustment Type](#). Refer to the base package's [C1-ADT-INFO](#) for an example. If such an algorithm is not plugged-in on the Adjustment Type, the system looks for a corresponding algorithm on the [installation record](#). Refer to the base package's [C1-ADI-INFO](#) for examples. If you prefer different formatting logic, your system administrator should configure the system appropriately.

Adjustment ID is the system-assigned, unique identifier of the adjustment.

Account ID is the account to which the adjustment is linked. The name of the main person on the account appears next to the account ID.

Use **Service Agreement ID** to define the service agreement whose value needs to be adjusted. Basic information about the service agreement appears adjacent.

Premise is a display-only field that shows the **Service Agreement**'s characteristic premise (i.e., the primary premise associated with the service agreement).

Indicate the **Adjustment Type**. This field is very important as it controls numerous aspects of the adjustment's impact on the customer's balance and your general ledger. This field is gray after the adjustment is frozen.

You can only choose certain adjustment types. The service agreement's SA type has a collection of valid adjustment profiles. You may only reference adjustment types that are listed in one of the adjustment type profiles linked to the SA type.

Enter the **Amount** of the adjustment. If the adjustment type is a calculated adjustment, the **Calculated Amount** and **Calculation Date** are displayed. The values of the calculated amount and calculation date are displayed after the adjustment is generated. The calculated amount shows the result of the generate adjustment algorithm. The calculation date is specified when you click Generate for a calculated adjustment type. It is used by the generate adjustment algorithm for any calculations that are effective dated (e.g., rate version or bill factor value).

Default note. The adjustment amount defaults based on the adjustment type. When you change the adjustment type, the amount changes accordingly. You may change the adjustment amount after it is defaulted.

On Behalf SA ID is a display-only field that is only populated on system-generated adjustments instigated by one customer, but attributed to another customer. For example, if chargeable field work is performed for a customer, but is payable by the customer's meter maintenance service provider, the adjustment will be linked to the service provider and will reference the customer's SA as its **On Behalf SA ID**. This field is only visible if it is populated on the adjustment.

You may not change an adjustment's **Adjustment Status** directly. Rather, you use the buttons in the **Adjustment Actions** area. Refer to [The Lifecycle Of An Adjustment](#) for the details.

If the status is **canceled**, the **Cancel Reason** is displayed.

Use the **Comments** to describe anything unusual about the adjustment.

The **Creation Date** defines the date on which the adjustment was created.

If the adjustment is subject to approval, a message indicating such appears. Clicking this message navigates the user to the **Approvals** tab. Refer to [The Big Picture of Adjustment Approvals](#) for more information.

The financial transaction (FT) grid contains the financial transactions associated with the adjustment. It only contains information after the adjustment is frozen. If the adjustment is canceled, a second row appears showing the details of the cancellation FT.

- **Financial Transaction ID** is the system-assigned unique identifier of the FT. Click the go to button to transfer to the financial transaction. On this page you can change certain aspects of the FT in question.
- **Arrears Date** is the date the FT starts aging.
- **Accounting Date** is the date the system uses to determine the FT's accounting period in your general ledger.
- **Current Amount** contains the FT's effect on the service agreement's current balance.
- **Payoff Amount** contains the FT's effect on the service agreement's payoff balance.
- **Bill ID** is the bill on which the FT appears (if it has been swept onto a bill). Click the adjacent go to button to transfer to the bill on which the FT appears. Note: an FT is linked to a bill the next time a bill is completed for the service agreement's account.

The **Calculation Lines** grid contains the details of the calculations associated with a [calculated adjustment](#). It only appears if the rate that calculated the adjustment amount created at least one calculation line. One row exists for every calculation involved in the process. This information is for audit purposes only and cannot be modified. The following information displays in the grid:

- If at least one of the calculation lines has characteristics, **Calc Line Char** displays go to buttons, allowing you to go to characteristics that are linked to a specific adjustment calculation line. This column is only displayed if at least one of the calculation lines has characteristics.
- **Sequence** is the system-assigned unique identifier of the calculation detail row.
- **Description on Bill** is the information about the calculation line that appears on the customer's bill.
- **Calculated Amount** is the calculated amount associated with the calculation line.
- The **Print** switch controls whether information about this line will print on the customer's bill.
- The **Appears in Summary** switch defines if this line's amount also appears on a summary line. This switch plays a part at bill print time – those lines that appear in a summary print in the left dollar column, those that don't appear in a summary print in the right dollar column.

Bill segments created by applying a rate have this switch turned on if the corresponding rate component is summarized on a summary rate component.

- **Unit of Measure (UOM)** is the unit of measure of the service quantity priced on the calculation line.
- **Time of Use (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 differs 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.
- **Rate Component Sequence** refers to the sequence number of the rate component on the applicable rate version that was used to calculate the line.
- **Measures 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.
- **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.
- **Description** describes the characteristic value that was used when the line's amount was calculated. This information is only displayed if the line was calculated using a bill factor (because only bill factors use characteristic values). Refer to [An Illustration Of A Bill Factor And Its Characteristics](#) for more information.

For more information, refer to [Calculated Adjustments](#).

The **Adjustment Actions** area contains buttons that you use to commit and cancel the adjustment's financial impact. The adjustment's status controls the button you can see and select. Refer to [The Lifecycle Of An Adjustment](#) for the details.

Freeze may not be enabled. Please be aware that you can prevent specific types of adjustments from being frozen until the account's next bill is **completed**. Refer to [Preventing SA Balances And The GL From Being Impacted Until Bill Completion](#) for how to enable this feature on your adjustment types.

Cancel may not be enabled. The cancel button is not enabled if the adjustment is linked to a bill that is written off.

Canceling An Accounts Payable (A/P) Adjustment. If you need to cancel an A/P adjustment (i.e. an adjustment of an adjustment type that has an A/P request type) that has already been extracted by A/P, refer to [How To Cancel An A/P Adjustment After It's Been Selected By A/P](#).

Adjustments - Characteristics

You use this page to link additional information to the adjustment. Open using **Financial, Adjustment, Characteristics**.

Description of Page

The characteristics collection contains information that describes miscellaneous information about the adjustment.

Characteristic Types. You can only choose characteristic types defined on the adjustment's [adjustment type](#).

The following fields display:

Characteristic Type	Indicate the type of characteristic.
Characteristic Value	Indicate the value of the characteristic.

Default Note. An adjustment's characteristics default from the [adjustment type](#).

Adjustments - Transfer Adjustment

The **Description of Page** section below describes the fields on this page. Refer to [How and When To Use An Adjustment](#) for instructions describing how to perform common maintenance functions.

The Transfer Adjustment is used to define the reciprocal adjustment associated with a transfer adjustment (there are always two adjustments associated with a transfer). Open using **Financial, Adjustment, Transfer Adjustment**.

Description of Page

Adjustment Info contains a concatenation of the adjustment amount, adjustment type and status.

Adjustment ID is the system-assigned, unique identifier of the adjustment.

Account ID is the account to which the adjustment is linked. The name of the main person on the account appears next to the account ID.

Use **Service Agreement ID** to define the transfer to service agreement. Basic information about the service agreement appears adjacent.

Premise is a display-only field that shows the **Service Agreement's** characteristic premise (i.e., the primary premise associated with the service agreement).

The **Adjustment Type** and adjustment **Amount** from the first page are displayed. If you need to change either value, return to the first page.

Adjustment Status shows the status of the transfer to adjustment. If you need to change the adjustment's status, use the action buttons.

Use the **Comments** to describe anything unusual about the adjustment.

Transfer Adj ID is the system-assigned, unique identifier of the adjustment.

The **Creation Date** defines the date on which the adjustment was created.

The financial transaction grid contains the financial transactions associated with the adjustment. It only contains information after the adjustment is frozen. If the adjustment is canceled, a second row appears showing the details of the cancellation FT.

- **Financial Transaction ID** is the system-assigned unique identifier of the FT. Click the go to button to transfer to the financial transaction. On this page you can change certain aspects of the FT in question.
- **Arrears Date** is the date the FT starts aging.
- **Accounting Date** is the date the system uses to determine the FT's accounting period in your general ledger.
- **Current Amount** contains the FT's effect on the service agreement's current balance.
- **Payoff Amount** contains the FT's effect on the service agreement's payoff balance.
- **Bill ID** is the bill on which the FT appears (if it has been swept onto a bill). Click the adjacent go to button to transfer to the bill on which the FT appears. Note: an FT is linked to a bill the next time a bill is completed for the account.

Adjustments - A/P Request

The **Description of Page** below describes the fields on this page. Refer to [How and When To Use An Adjustment](#) for instructions describing how to perform common maintenance functions.

The A/P Request page contains information about adjustments used to refund money via an A/P check request. This page is only relevant if:

- The adjustment's adjustment type references an A/P Request Type (i.e., it will be interfaced to your accounts payable system which will cut the check), AND
- The adjustment is **frozen**.

Open this page using **Financial, Adjustment, A/P Request**.

Description of Page

Adjustment Info contains a concatenation of the adjustment amount, adjustment type and status.

Adjustment ID is the system-assigned, unique identifier of the adjustment.

Account ID is the account to which the adjustment is linked.

Adjustment type controls A/P check requests. The A/P Request information is populated when an adjustment used to refund money via an A/P check request is **frozen**. Whether or not an adjustment is interfaced to A/P is controlled by the adjustment's adjustment type. The adjustment type's A/P Request Type controls the payment date and the bank.

Name is the name printed on the check. This name is derived from the account's main person:

- If this person has an override mailing name, the first line of the override mailing name is used.
- If this person does not have an override mailing name, the person's primary name is used.

Payment Selection Status is the status of the check request. The values are:

- **Not Selected for Payment** before it's selected by A/P for payment

- **Requested for Payment** after it's selected by A/P
- **Paid** after it's paid by A/P
- **Canceled** if it's been canceled in A/P
- **Hold** if it's been held in A/P

Scheduled to Pay is the date on which the check is scheduled to be cut. This is equal to the adjustment date plus the Due Days on the adjustment type's A/P request type.

The following fields are provided in order to support a two-way interface with an A/P system. The delivered system does not have processes which update these fields.

- **Payment Date** is the date on which the check was cut in A/P. This field is only populated after A/P cuts the check.
- **Paid Amount** is the amount of the check. This field is only populated after A/P cuts the check.
- **A/P Request ID** is the system-assigned, unique identifier of the A/P check request.
- **Payment Number** is the system-assigned number of the payment in A/P (this number typically appears on the printed check). This field is only populated after A/P cuts the check.

The address information in the bottom frame is the address to which the check is mailed. This is the billing address of the main customer linked to the account. After the adjustment is frozen, this information is automatically populated based on the address information for the account. This information is modifiable until the status is **Paid**.

Adjustments - Approval

This page only appears after an adjustment has started the approval process. Open this page using **Financial, Adjustment, Approval**.

Refer to [The Big Picture of Adjustment Approvals](#) for more information about the approval process.

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

Contents

- [Adjustment Information](#)
- [Approval Request](#)
- [Approval Request Log](#)

Adjustment Information

The Adjustment Information zone contains display-only information about the adjustment.

Approval Request

The Approval Request zone shows the current and future approvers of an adjustment. This zone only appears if the adjustment is in the approval process.

If the current user has approval authority, the following functions are available:

- Click the **Approve** button to approve the adjustment.
- Click the **Reject** button to reject the adjustment.

Approval Request Log

The Approval Request Log zone contains the history of the adjustment's approval.

Financial - Adjustment Calculation Line Characteristics

This page displays the characteristics that are linked to a specific adjustment calculation line. The information on this page is for audit purposes only and cannot be changed.

Open this page by clicking on the characteristics go to button on the calculation lines grid of the [Adjustments - Main](#) page.

Description of Page

Account is the account to which the adjustment is linked. The name of the main person on the account appears next to the account ID.

The **SA** is the service agreement whose value needs to be adjusted. Basic information about the service agreement appears adjacent.

Premise shows the service agreement's characteristic premise (i.e., the primary premise associated with the service agreement).

The **Adjustment Info** displays information to identify the adjustment, including amount, type, and status.

Adjustment ID displays the identification of the adjustment, and **Calc Line** displays the calculation line with which the characteristics are associated.

The **Description on Bill** displays the description of the calculation line that appears on the customer's bill.

The characteristics grid displays the **Characteristic Types** and the **Characteristic Values** associated with the calculation line.

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

How and When To Use An Adjustment

A last resort. Adjustments should be viewed as a last resort because you should correct the cause of errors, rather than the effect. For example, if a customer's bill is incorrect, you should fix the customer's rate / meter reads / contract terms and then perform a cancel / rebill. Issuing an adjustment to fudge the balance is really just that - a fudge.

The topics in this section describe how to perform common adjustment maintenance functions.

Contents

[How To Create A Transfer Adjustment](#)
[How To Create A Calculated Adjustment](#)
[How To Cancel An A/P Adjustment After It Has Been Selected By A/P](#)
[How To Correct Normal Service Agreement Debt](#)
[How To Apply Ad Hoc Fees To A Service Agreement](#)
[How To Apply Interest To A Deposit](#)
[How To Apply A Deposit To Outstanding Debt](#)
[How To Refund A Deposit With A Check](#)
[How To Write-Down Deposit Due](#)
[How To Refund A Credit Balance With A Check](#)
[How To Transfer A Credit Balance To A Service Agreement In Arrears](#)
[How To Change The Age Of Debt](#)
[How To Write-Down A Charitable Contribution Service Agreement Debt](#)
[How To Write-Off Debt](#)
[How To Cancel A Service Agreement's Budget](#)
[How To Adjust A Service Agreement On A Budget](#)
[How To Adjust An Installment Plan Service Agreement](#)
[How To Use An Adjustment To Change The GL Distribution](#)

How To Create A Transfer Adjustment

The following steps describe how to create a transfer adjustment on the Adjustment page.

Note. You cannot use an adjustment type with a calculated amount for transfer adjustments.

Warning! These steps only make sense in the context of the page used to maintain Adjustments. Refer to [Adjustments - Main Information](#) for the details.

To create a transfer adjustment:

- Specify the “transfer from” account and service agreement on the Adjustment – Main page.
- Specify the adjustment type and the amount and transfer to the Transfer Adjustment page.
- Specify the “transfer to” service agreement and click the **Generate** and **Freeze** buttons to generate and freeze the adjustments’ financial transactions. Please be aware that the **Freeze** button may be disabled for certain types of adjustments. This is because you can prevent specific types of adjustments from being frozen until the account’s next bill is **completed**. Refer to [Preventing SA Balances And The GL From Being Impacted Until Bill Completion](#) for how to enable this feature on your adjustment types.

How To Create A Calculated Adjustment

The following steps describe how to create a [calculated adjustment](#) on the Adjustment page.

Warning! These steps only make sense in the context of the page used to maintain Adjustments. Refer to [Adjustments - Main Information](#) for the details.

To create a calculated adjustment:

1. Open **Financial, Adjustment, Main** and specify the **Service Agreement** to which the adjustment will be applied.

2. Specify the **Adjustment Type** that calculates the amount, and enter a base **Amount**.
3. Click **Generate**. The **Generate** dialog box opens.
4. In the resulting dialog box, specify an **Accounting Date** and **Generation Reference Date**. Click **Calculate**.
5. Click **Freeze** to freeze the calculated transactions.

The **Calculation Date** that you supply is used by the generate adjustment algorithm for any calculations that are effective dated (e.g., rate version or bill factor value).

Please be aware that the **Freeze** button may be disabled for certain types of adjustments. This is because you can prevent specific types of adjustments from being frozen until the account's next bill is **completed**. Refer to [Preventing SA Balances And The GL From Being Impacted Until Bill Completion](#) for how to enable this feature on your adjustment types.

How To Cancel An A/P Adjustment After It Has Been Selected By A/P

Adjustments that are interfaced to A/P (because A/P needs to cut a refund check) sometimes need to be canceled. You may cancel an A/P adjustment in the CIS system while its A/P request is **Not Selected**. If the A/P request is **Requested for Payment** or **Paid** you must first cancel the payment in A/P. Canceling the payment in A/P changes the A/P request in CIS to **Canceled**. At this point, you can cancel the adjustment in CIS.

How To Correct Normal Service Agreement Debt

If a customer is unhappy with the balance on a "normal" service agreement and you cannot fix the source of their unhappiness (i.e., the bills and payments), create an adjustment. Simply select an adjustment type that describes why you are adjusting the balance and enter the amount of the adjustment (a negative number if you are crediting the customer).

What's normal? There really is no such thing as a "normal" service agreement. We use this term to reference service agreements used to bill for services like utility service, land leases, home warranties, security monitoring, etc. where the current amount is the same as the payoff amount.

Warning! A service agreement that is part of a budget plan is NOT normal. Refer to [How To Adjust A Service Agreement On A Budget](#) for more information.

How To Apply Ad Hoc Fees To A Service Agreement

If you need to apply a fee (e.g., late payment charge, connection charge, NSF charge) to a "normal" service agreement, issue an adjustment using the appropriate adjustment type. The adjustment amount should be a positive number so that the customer's balance increases.

Warning! Take care if you apply a fee to a service agreement that's on a budget. The adjustment affects current balance and payoff balance by the amount of the fee. If you didn't mean to affect the customer's current balance, you'll have to issue another adjustment to change it.

How To Apply Interest To A Deposit

You should only have to issue adjustments to calculate and refund interest if you need to override the system's automated procedures. Refer to [Controls Interest Calculation](#) for the details.

If you need to manually apply interest to a deposit, issue an adjustment using the appropriate adjustment type. The adjustment amount should be a negative number (you are crediting the customer's deposit balance).

If you want to apply the interest to the customer's outstanding balance, refer to [How To Apply A Deposit To Outstanding Debt](#).

If you want to cut a check for the deposit, refer to [How To Refund A Deposit With A Check](#).

How To Apply A Deposit To Outstanding Debt

You should only have to issue adjustments to apply a deposit to outstanding debt if you need to override the system's automated procedures. Refer to [Refunding Deposits](#) for the details.

Applying a deposit is a two-step process:

- The deposit service agreement's payoff balance contains the amount of deposit you are holding on behalf of the customer's debt. Its current balance should be zero (i.e., the customer doesn't owe any additional deposit). Before you apply the deposit balance, we recommend making the deposit's current balance equal to the amount to be transferred. You do this by issuing an adjustment that just affects the deposit's current balance. The amount of this adjustment should be a negative number.
- After the deposit's current balance reflects an appropriate credit amount (equal to the amount to be applied to the customer's debt), use the Transfer Adjustment process to apply the deposit's credit balance to another service agreement's debit balance. The deposit service agreement is the transfer from service agreement; the delinquent service agreement is the transfer to service agreement. The adjustment amount should be a positive number. Refer to [How To Create A Transfer Adjustment](#) for more information.

Warning! Take care if you apply a deposit to a service agreement that's on a budget. The transfer adjustment affects current balance and payoff balance by the same amount. If you didn't mean to affect the customer's current balance, you'll have to issue another adjustment to change it.

How To Refund A Deposit With A Check

You should only have to issue adjustments to apply a deposit to outstanding debt if you need to override the system's automated procedures. Refer to [Refunding Deposits](#) for the details.

Refunding a deposit with a check (cut by your A/P system) is a two step process:

- The deposit service agreement's payoff balance contains the amount of deposit you are holding on behalf of the customer's debt. Its current balance should be zero (i.e., the customer doesn't owe any additional deposit). Before you refund the deposit balance, we

recommend making the deposit's current balance equal to the amount to be refunded. You do this by issuing an adjustment that just affects the deposit's current balance. The amount of this adjustment should be a negative number.

- After the deposit's current balance reflects an appropriate credit amount (equal to the amount to be refunded), issue a refund adjustment. The adjustment amount should be a positive number. This adjustment causes a check request to be interfaced to your A/P system.

How To Write-Down Deposit Due

If a customer doesn't pay their deposit and you don't expect them to, create an adjustment to write-down the current balance. Simply select an adjustment type that describes why you are adjusting the balance and enter the amount of the adjustment (a negative number).

How To Refund A Credit Balance With A Check

If you need to refund a credit balance on a "normal" service agreement, issue a refund adjustment using the appropriate adjustment type. The adjustment amount should be a positive number. This adjustment causes a check request to be interfaced to your A/P system.

Warning! Take care if you refund a credit balance from a service agreement that's on a budget. The adjustment affects current balance and payoff balance by the amount of the refund. If you didn't mean to affect the customer's current balance, you'll have to issue another adjustment to change it.

How To Transfer A Credit Balance To A Service Agreement In Arrears

Use the Transfer Adjustment process to apply a "normal" service agreement's credit balance to another service agreement's debit balance. The credit service agreement is the transfer from service agreement; the delinquent service agreement is the transfer to service agreement. The adjustment amount should be a positive number. Refer to [How To Create A Transfer Adjustment](#) for more information.

Warning! Take care if you apply a credit to a service agreement that's on a budget. The transfer adjustment affects current balance and payoff balance by the transfer amount. If you didn't mean to affect the customer's current balance, you'll have to issue another adjustment to change it.

How To Change The Age Of Debt

If a service agreement's debt is older (or younger) than what it should be (for whatever reason), you should do the following.

- Use the Transfer Adjustment process to re-age the debt. The transfer from and transfer to service agreements will be the same. The adjustment amount should be a negative number. Refer to [How To Create A Transfer Adjustment](#) for more information.
- After creating the transfer adjustment, drill into the transfer to adjustment and populate the arrears date with the day on which the debt should start aging (e.g., if the debt should be 7

days old, the arrears date should be the current date minus 7 days).

Warning! Be careful if you transfer debt from a service agreement that's on a budget. Before doing the transfer, you should make the customer's current balance equal their payoff balance.

How To Write-Down A Charitable Contribution Service Agreement Debt

If a customer doesn't want to contribute as much to the charity as the agreed to, create an adjustment to fix the current balance (charity service agreements shouldn't have a payoff balance as the "debt" is voluntary). Simply select an adjustment type that describes why you are adjusting the balance and enter the amount of the adjustment (a negative number).

How To Write-Off Debt

To write-off debt:

Warning! You don't have to do the tasks outlined below if you use the [Write Off - Main](#) page. This page does these tasks for you.

- Create a write-off service agreement using the Start Service page (you could also use the Service Agreement page to do this).
- Use the Transfer Adjustment process to transfer the delinquent debt to the write-off service agreement. The original service agreement is the transfer from service agreement; the write-off service agreement is the transfer to service agreement. The adjustment amount should be a negative number. Refer to [Transfer Adjustments](#) for more information.

Warning! Take care if you transfer debt from a service agreement that's on a budget. Before doing the transfer, you should make the service agreement's current balance equal their payoff balance.

How To Cancel A Service Agreement's Budget

Warning! You don't have to do the tasks outlined below if you use the [Account – Budget](#) panel. This panel does these tasks for you.

Canceling a service agreement's budget involves the following steps:

- Update the service agreement to indicate the service agreement is no longer on a budget.
- Issue an adjustment to make the service agreement's current balance equal the payoff balance.

Credit and collections (C&C). You can change the arrears date on the adjustment that synchronizes current and payoff balance to reflect the true age of the debt. You do this on the financial transaction associated with the adjustment. If you do this, C&C will monitor the old debt accordingly.

How To Adjust A Service Agreement On A Budget

If you need to adjust the debt on a service agreement that's on a budget plan, you have to ask yourself why you're adjusting the balance:

- If you're adjusting the balance to correct the payoff balance (for whatever reason), choose an adjustment type that just affects this balance.
- If you're adjusting the balance to correct the current balance (for whatever reason), choose an adjustment type that just affects this balance.
- If you need to adjust both balances equally, choose an adjustment type that affects both.

How To Adjust An Installment Plan Service Agreement

If you need to adjust the debt on a service agreement that's used to bill a total amount in installments, you have to ask yourself why you're adjusting the balance:

- If you're adjusting the balance to correct the payoff balance (for whatever reason), choose an adjustment type that just affects this balance.
- If you're adjusting the balance to correct the current balance (for whatever reason), choose an adjustment type that just affects this balance.

If you need to adjust both balances equally, choose an adjustment type that affects both.

How To Use An Adjustment To Change The GL Distribution

Assume you have a frozen bill segment that affected an incorrect GL distribution code (e.g., it booked revenue to commercial – time-of-use – revenue and it should have booked it to industrial – time-of-use – revenue. You have two ways to correct such a situation:

- You could correct the cause of the incorrect distribution code (probably incorrect GL information on a rate component) and cancel / rebill the bill segment. Refer to [Cancel / Rebill](#) for more information about this method.
- You could create an adjustment that does not impact the customer's balance but does have a GL entry. This GL entry would reverse the effect of the original distribution code and have an offsetting entry to the correct distribution code. The following points explain how to do this:
 - Choose an adjustment type that only impacts the GL (i.e., the adjustment's adjustment type has an FT algorithm that doesn't impact the customer's current or payoff balance, it only impacts the GL).
 - Enter an adjustment amount of zero.
 - Generate the adjustment.
 - Drill into the adjustment's [financial transaction](#) and enter two (or more) GL distribution rows that reflect the redistribution of the amounts.
 - Return to the adjustment and freeze it.

Adjusting statistical amounts. You can use the above approach to change the statistical amount that is interfaced to the general ledger. The statistical amount is simply another column on the financial transaction's GL distribution rows.

Interfacing Adjustments From External Sources

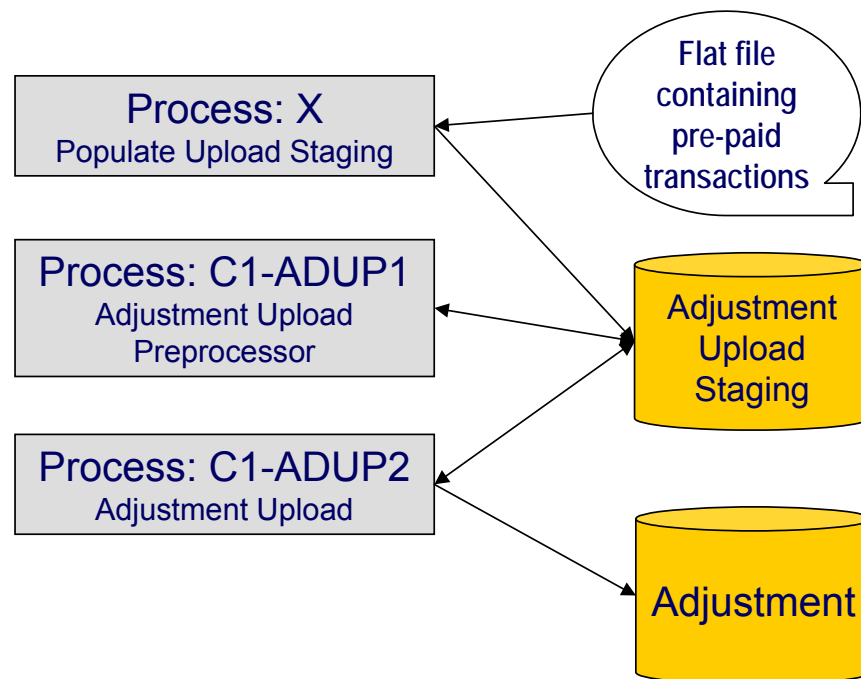
The topics in this section describe how adjustments are uploaded from an external source.

Contents

- [Interfacing Adjustments](#)
- [Suspense Adjustments](#)
- [Maintaining Adjustment Staging Control](#)
- [Maintaining Adjustment Upload Staging](#)

Interfacing Adjustments

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



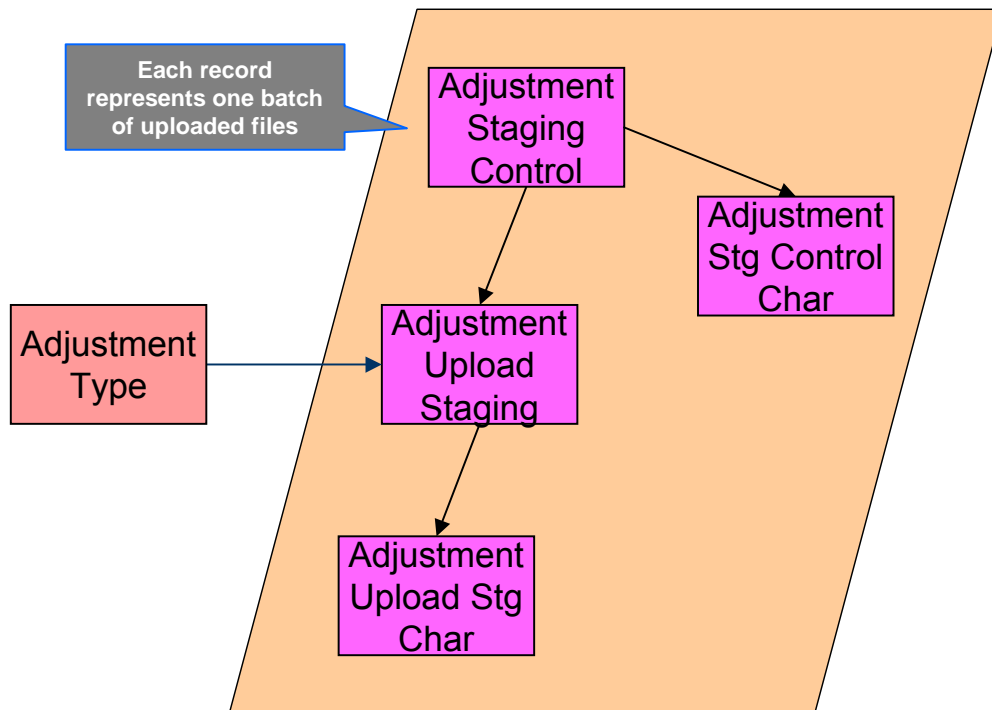
The topics in this section describe how these processes work.

Contents

- [Process X - Populate Adjustment Upload Records](#)
- [Process C1-ADUP1 - Preprocess Adjustment Uploads](#)
- [Process C1-ADUP2 - Upload Adjustments](#)

Process X - Populate Adjustment Upload Records

Process X refers to the mechanism used by your organization to populate the various staging tables (shown in the orange section of the following ERD).



The topics in this section describe each of these tables.

Contents

- [Adjustment Staging Control](#)
- [Adjustment Staging Control Characteristic](#)
- [Adjustment Upload Staging](#)
- [Adjustment Characteristic Upload Staging](#)
- [The Lifecycle of an Adjustment Staging Control Record](#)
- [The Lifecycle of an Adjustment Upload Staging Record](#)

Adjustment Staging Control

You must create an adjustment staging control record for each batch of adjustments to be uploaded into the system. The name of this table is CI_ADJ_STG_CTL. The following describes each column on this table.

Column Name	Length	Req'd	Data Type	Comments
ADJ_STG_CTL_ID	12	Y	N	This is the unique identifier of the adjustment staging control record. This key is a sequential number so you can use a database function to assign the value when populating the table.
CRE_DTTM	15	Y	DateTime	This is the date / time on which the adjustment staging control record was created. This must be

				populated with the current date / time.
ADJ_STG_CTL_STATUS_FLG	4	Y	A/N	This must be set to <i>P</i> for <i>Pending</i> .
ADJ_STG_UP_REC_CNT	10	Y	N	This is the total number of adjustment upload staging records that are linked to this adjustment staging control.
TOT_ADJ_AMT	13.2	Y	N	This column must equal the sum of adjustment amounts on the adjustment upload staging records that are linked to this adjustment staging control.
CURRENCY_CD	3	Y	A/N	This must be a valid currency code in the system. Refer to Defining Currency Codes for more information.
MESSAGE_CATEGORY	5	N	N	Leave this blank. The adjustment upload preprocessor populates this when an error occurs during upload.
MESSAGE_NBR	5	N	N	Leave this blank. The adjustment upload preprocessor populates this when an error occurs during upload.

Adjustment Staging Control Characteristic

You must create an adjustment staging control characteristic record for each characteristic that you would like to link to the adjustment staging control. The name of this table is CI_ADJ_STG_CTL_CHAR. The following describes each column on this table.

Column Name	Length	Req'd	Data Type	Comments
ADJ_STG_CTL_ID	12	Y	N	This must correspond with the prime key of the related CI_ADJ_STG_CTL record.
CHAR_TYPE_CD	8	Y	A/N	This must correspond with a characteristic type that is defined as valid for <i>adjustment staging control</i> . Refer to Setting Up Characteristic Types & Their Values for more information.
SEQ_NUM	3	Y	N	This should be set to <i>10</i> unless you have multiple values for a given adjustment staging control and characteristic type.
CHAR_VAL	16	N	A/N	Populate this field if your characteristic type is <i>predefined</i> .
ADHOC_CHAR_VAL	254	N	A/N	Populate this field if your characteristic type is <i>ad-hoc</i> or <i>file</i> .

				<i>location.</i>
CHAR_VAL_FK1 - CHAR_VAL_FK5	50 each	N	A/N	Populate these fields if your characteristic type is <i>foreign key reference</i> . Up to five columns of 50 bytes each are provided to accommodate compound keys.

Adjustment Upload Staging

You must create an adjustment upload staging record for each adjustment you want to upload. The name of this table is CI_ADJ_STG_UP. The following describes each column on this table.

Column Name	Length	Req'd	Data Type	Comments
ADJ_STG_UP_ID	12	Y	N	This is the unique identifier of the adjustment upload staging record. This key is a sequential number so you can use a database function to assign the value when populating the table.
ADJ_STG_CTL_ID	12	Y	N	The ID of the adjustment staging control that is linked to this adjustment upload staging record.
ADJ_TYPE_CD	8	Y	A/N	This must correspond to the prime key of one of your adjustment types. Refer to Setting up Adjustment Types for more information.
ADJ_STG_UP_STATUS_FLG	4	Y	A/N	This must be set to <i>P</i> for <i>Pending</i> .
CREATE_DT	10	Y	Date	The date when the adjustment occurred.
ADJ_AMT	13.2	Y	N	The amount of the adjustment.
ADJ_SUSPENSE_FLG	4	Y	N	This must be set to <i>NSUS</i> for <i>Not In Suspense</i> .
SA_ID	10	N	A/N	This must correspond to a valid service agreement in the system. If you leave this blank and opt to let the system find the SA during adjustment upload preprocessing, a <i>Determine SA</i> algorithm must be plugged in on the associated adjustment type. This plug-in is meant to derive the SA ID based on supplied miscellaneous information – e.g. information supplied via adjustment characteristic upload staging.
ADJ_ID	10	N	A/N	Leave this blank. The adjustment upload process populates this.

SUSPENSE_ADJ_ID	10	N	A/N	Leave this blank. The adjustment upload process populates this.
MESSAGE_CATEGORY	5	N	N	Leave this blank. The adjustment upload background processes populate this when an error occurs during upload.
MESSAGE_NBR	5	N	N	Leave this blank. The adjustment upload background processes populate this when an error occurs during upload.

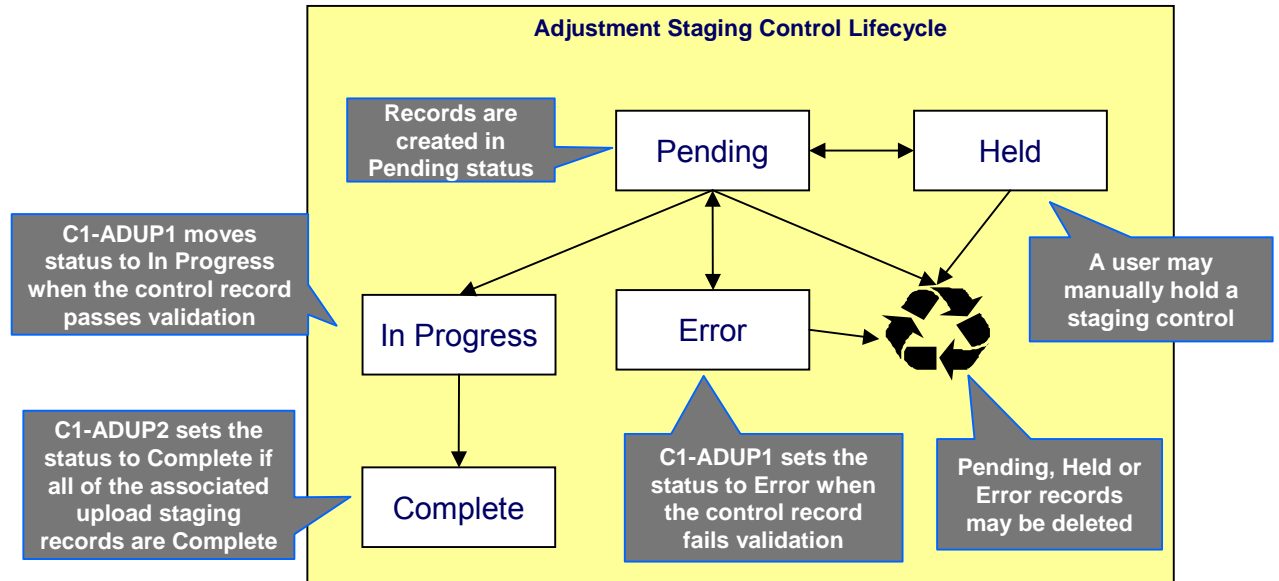
Adjustment Characteristic Upload Staging

You must create an adjustment characteristic upload staging record for each characteristic that you would like to link to the adjustment upload staging. The name of this table is CI_ADJ_STG_UP_CHAR. The following describes each column on this table.

Column Name	Length	Req'd	Data Type	Comments
ADJ_STG_UP_ID	12	Y	N	This must correspond with the prime key of the related CI_ADJ_STG_UP record.
CHAR_TYPE_CD	8	Y	A/N	This must correspond with a characteristic type that is defined as valid for <i>adjustment</i> . Refer to Setting Up Characteristic Types & Their Values for more information.
SEQ_NUM	3	Y	N	This should be set to 10 unless you have multiple values for a given adjustment upload staging and characteristic type.
CHAR_VAL	16	N	A/N	Populate this field if your characteristic type is <i>predefined</i> .
ADHOC_CHAR_VAL	254	N	A/N	Populate this field if your characteristic type is <i>ad-hoc</i> or <i>file location</i> .
CHAR_VAL_FK1 - CHAR_VAL_FK5	50 each	N	A/N	Populate these fields if your characteristic type is <i>foreign key reference</i> . Up to five columns of 50 bytes each are provided to accommodate compound keys.

The Lifecycle of an Adjustment Staging Control Record

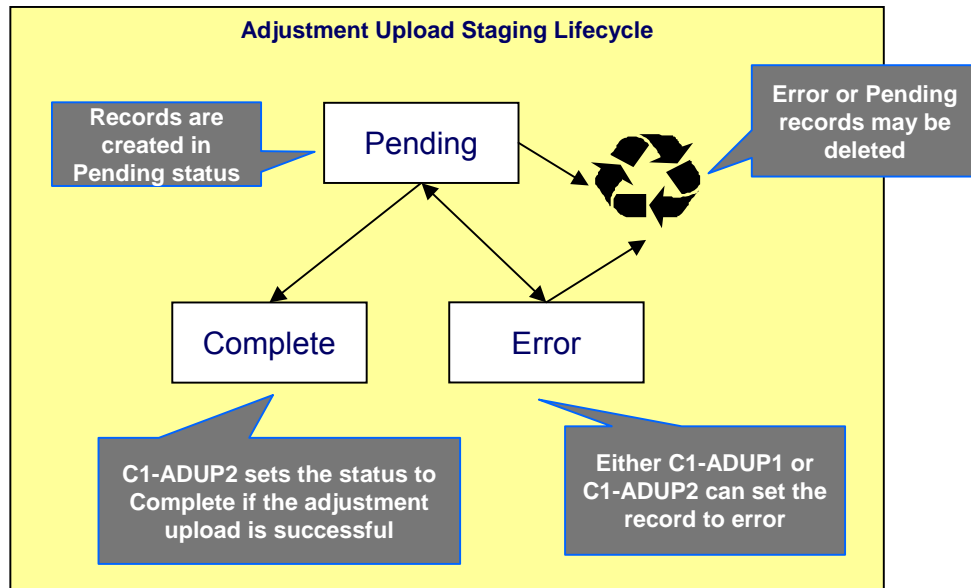
The following diagram shows the possible lifecycle of an adjustment staging control record.



- **Pending.** An adjustment staging control record is created in this state. The **C1-ADUP1** process selects **pending** adjustment staging control records for validation.
- **In Progress.** The **C1-ADUP1** process sets a **pending** record to **in progress** when the totals on the adjustment staging control are successfully validated against the totals from the associated adjustment upload staging records.
- **Complete.** The **C1-ADUP2** process sets the adjustment staging control's status to **complete** when all adjustment upload staging records linked to the adjustment staging control are **complete**.
- **Error.** The **C1-ADUP1** process sets a **pending** record to **error** if the adjustment staging control fails validation. The status may be set back to **pending** after the error is fixed.
- **Held.** The **held** status is available for situations where you want to prevent or delay the upload of a batch of adjustment staging records. The status may be set back to **pending** when the batch of records is ready for upload.
- **Pending, error** and **held** records may be deleted.

The Lifecycle of an Adjustment Upload Staging Record

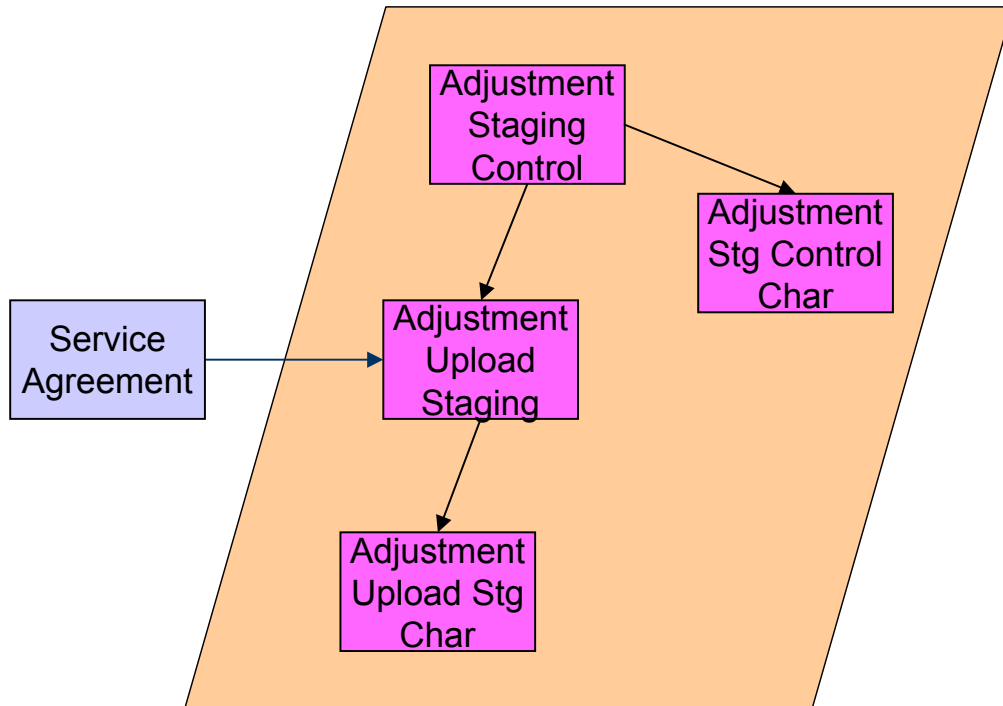
The following diagram shows the possible lifecycle of an adjustment upload staging record.



- **Pending.** Adjustment upload records are created in this state. The **C1-ADUP2** process selects **pending** adjustment upload records and creates adjustments for each of them.
- **Complete.** The **C1-ADUP2** process sets a **pending** record to **complete** if an adjustment is successfully created.
- **Error.** Either **C1-ADUP1** or **C1-ADUP2** process sets **pending** record to **error** when it encounters an error during the upload process. The status may be set back to **pending** after the error is fixed.
- **Pending** and **error** records may be deleted.

Process C1-ADUP1 - Preprocess Adjustment Uploads

The batch process identified by batch process ID **C1-ADUP1** refers to the background process that validates adjustment staging control records and populates service agreement IDs on adjustment upload staging records that do not specify an SA ID.



Contents

Phase 1 - Validate Adjustment Staging Controls

Phase 2 - Populate SA ID

Phase 1 - Validate Adjustment Staging Controls

The following points describe, at a high level, the first phase of the adjustment upload pre-process:

- For each **Pending** adjustment staging control,
 - Check that the record count on the adjustment staging control record equals the number of adjustment upload staging records that are linked to the adjustment staging control.
 - Check that the total adjustment amount on the adjustment staging control record equals the sum of the adjustment amounts from the adjustment upload staging records that are linked to the adjustment staging control.
 - If the adjustment staging control passes validation, set its status to **In Progress**. Otherwise, set the status to **Error**. Create a To Do entry using the inputs To Do type and To Do role (if supplied) for adjustment staging control errors. (Complete any outstanding To Do entries for the adjustment staging control before creating a new To Do entry.)
 - If no errors occur, perform To Do cleanup by completing any outstanding To Do entries that were previously created for the adjustment staging control.

Note. You can fix errors by going to the To Do entry and drilling into the adjustment staging control page. Don't forget to change the adjustment staging control's status back to **Pending** after fixing the error.

Phase 2 - Populate SA ID

The following points describe, at a high level, the second phase of the adjustment upload pre-process:

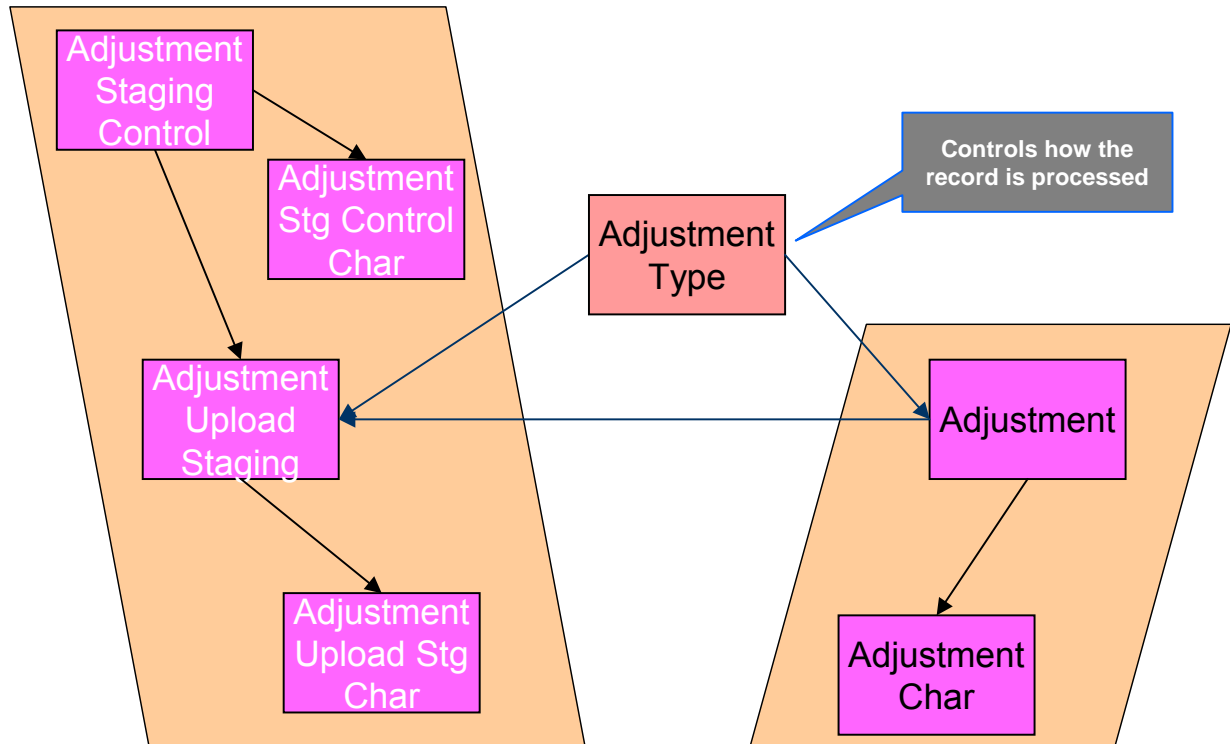
- For each **Pending** adjustment upload staging that is linked to an **In Progress** adjustment staging control AND does not have the SA ID,
 - Execute the **Determine SA** algorithm that is plugged in on the [adjustment type](#).
 - If the algorithm returns a valid SA ID, stamp that SA ID onto the adjustment upload staging. If the algorithm also returns an indication that the adjustment is to be put into suspense, set the suspense flag on the adjustment upload staging to **In Suspense**. Refer to [Suspense Adjustments](#) for more information on how suspense adjustments are handled.
 - If the algorithm returns an error, set the adjustment upload staging record's status to **Error**. Create a To Do entry using the inputs To Do type and To Do role (if supplied) for adjustment upload staging errors. (Complete any outstanding To Do entries for the adjustment upload staging before creating a new To Do entry.)
 - If no errors occur, perform To Do cleanup by completing any outstanding To Do entries that were previously created for the adjustment upload staging.

Note. You can fix errors by going to the To Do entry and drilling into the adjustment upload staging page. Don't forget to change the adjustment upload staging's status back to **Pending** after fixing the error.

Process C1-ADUP2 - Upload Adjustments

The batch process identified by batch process ID **C1-ADUP2** refers to the background process that creates adjustments for all adjustment upload staging records that are stamped with an SA ID.

The following diagram and section describe, at a high level, the processing done in the **C1-ADUP2** background process.



- For each **Pending** adjustment upload staging that has an SA ID,
 - Add a frozen adjustment using the amount, creation date and adjustment type on the staging record.

Note. If the type of adjustment being uploaded is one that is calculated, the algorithm that's plugged in on the adjustment type will calculate the adjustment amount and generate associated calculation lines (if applicable). See [Calculated Adjustments](#) for more information.

- If adjustment creation is successful, update the adjustment ID on the staging record and set the staging record's status to **Complete**.
- If adjustment creation results in an error, set the staging record's status to **Error**. Create a To Do entry using the To Do type on which this batch process (**C1-ADUP2**) is defined as creation process. (Complete any outstanding To Do entries for the adjustment upload staging before creating a new To Do entry.)
- If no errors occur, perform To Do cleanup by completing any outstanding To Do entries that were previously created for the adjustment upload staging.

Suspense Adjustments

Contents

[What Are Suspense Adjustments?](#)

[How Are Suspense Adjustments Resolved?](#)

What Are Suspense Adjustments?

When the adjustment upload preprocessor is unable to identify a valid SA ID, it can post the adjustment to a suspense SA. This is similar to how a payment is posted to a suspense account when a valid account ID could not be identified during payment upload.

Putting adjustments in suspense is optional. Therefore, this logic sits in a plug-in spot. If suspense adjustments are applicable to you, you must plug in your suspense logic in a **Determine SA** algorithm on [Adjustment Type](#).

How Are Suspense Adjustments Resolved?

A background process can be run periodically to automatically resolve suspense adjustments. Since rules for resolving suspense may vary, this logic sits in a plug-in spot. You must plug in your resolve logic in a **Resolve Suspense** algorithm on [Adjustment Type](#).

Process C1-ADURS - Resolve Suspense Adjustments

The batch process identified by batch process ID **C1-ADURS** refers to the background process that resolves suspense adjustments.

For each adjustment upload staging that is **In Suspense**, execute the **Resolve Suspense** algorithm that is plugged in on the [adjustment type](#).

Maintaining Adjustment Staging Control

Use this page to add, view or modify an adjustment staging control record. Open this page using **Financial, Adjustment Staging Control**.

Description of Page

Adjustment Staging Control ID is the unique identifier of the adjustment staging control record.

Create Date/Time is when the adjustment staging control was created. This field is protected after the record is saved.

Number of Staging Records is the number of adjustment upload staging records that are linked to this adjustment staging control record.

Total Adjustment Amount is the total of adjustment amounts from the adjustment upload staging records that are linked to this adjustment staging control record.

Enter the **Currency** for the total adjustment amount on the adjustment staging control. This is also the currency of the adjustment amounts on the associated adjustment upload staging records.

Status shows the state of the adjustment staging control. Possible values are: **Pending**, **In Progress**, **Complete**, **Error**, and **Held**.

A **Hold** button appears adjacent to the status description if **Status** is **Pending**. Click this button to set the status to **Held**. The adjustment upload process will ignore staging controls in this state.

A **Pend** button appears adjacent to the status description if **Status** is either **Held** or **Error**. Click this button to set the status to **Pending**. An error message box also appears below the status description if **Status** is **Error**.

Use the **Characteristics** collection to capture miscellaneous information about an adjustment staging control.

- **Characteristic Type** The type of characteristic.
- **Characteristics Value** The value of the characteristic.

Note. You can only choose characteristic types defined as permissible on an adjustment staging control. Refer to [Setting Up Characteristic Types & Their Values](#) for more information.

Maintaining Adjustment Upload Staging

Use this page to add, view or modify an adjustment upload staging record. Open this page using **Financial, Adjustment Upload Staging**.

Description of Page

Adjustment Upload ID is the unique identifier of the adjustment upload staging record.

Adjustment Staging Control ID is the identifier of the adjustment staging control to which the adjustment upload staging is linked.

Indicate the **Adjustment Type**. This field is very important as it controls numerous aspects of the adjustment's impact on the customer's balance and your general ledger.

You can only choose certain adjustment types. The service agreement's SA type has a collection of valid adjustment profiles. You may only reference adjustment types that are listed in one of the adjustment type profiles linked to the SA type.

Enter an **Adjustment Amount**.

Creation Date is the date when the adjustment occurred. This will be the creation date on the uploaded adjustment. This field is protected after the record is saved.

Service Agreement indicates the SA to which the adjustment was posted. If the adjustment is in suspense, the label shows **Suspense SA** instead.

Adjustment is a reference to the adjustment that got created from the upload.

Status shows the state of the adjustment upload staging. Possible values are: **Pending**, **Complete**, and **Error**.

If **Status** is **Error**, a **Pend** button appears adjacent to the status description. In addition, an error message box displays below the status description.

If the adjustment is in suspense, a message appears to indicate the condition and the **Suspense Adjustment** is shown. A different message appears when the suspense is resolved.

Use the **Characteristics** collection to keep miscellaneous information about an adjustment upload staging.

- **Characteristic Type** The type of characteristic.
- **Characteristics Value** The value of the characteristic.

Default Note. An adjustment upload staging's characteristics default from the [adjustment type](#).

Credit & Collections

The system periodically monitors how much your customers owe to ensure they haven't violated your collection criteria. When a violation is detected, the system generates the appropriate events (e.g., letters, disconnect field activities, To Do entries, write-off outstanding debt, etc.). This section describes how to manage your credit & collections processing.

In the section, [The Big Picture Of Credit & Collections \(C&C\)](#), we describe how to set up the control tables that automate most of your credit & collections processing. If you set these tables up properly, the system will manage most of your collection, severance and write-off processes; little or no user intervention should be necessary. You will only access the C&C pages when you need to override what the system does.

Contents

- [The Big Picture Of Credit & Collections](#)
- [Collection Process Maintenance](#)
- [Severance Process Maintenance](#)
- [Write-off Process Maintenance](#)
- [Writing Off Uncollectable Debt](#)
- [Collection Referral](#)
- [Payment Arrangements and Pay Plans](#)

The Big Picture Of Credit & Collections

The topics in this section provide background information about a variety of credit & collections issues.

Contents

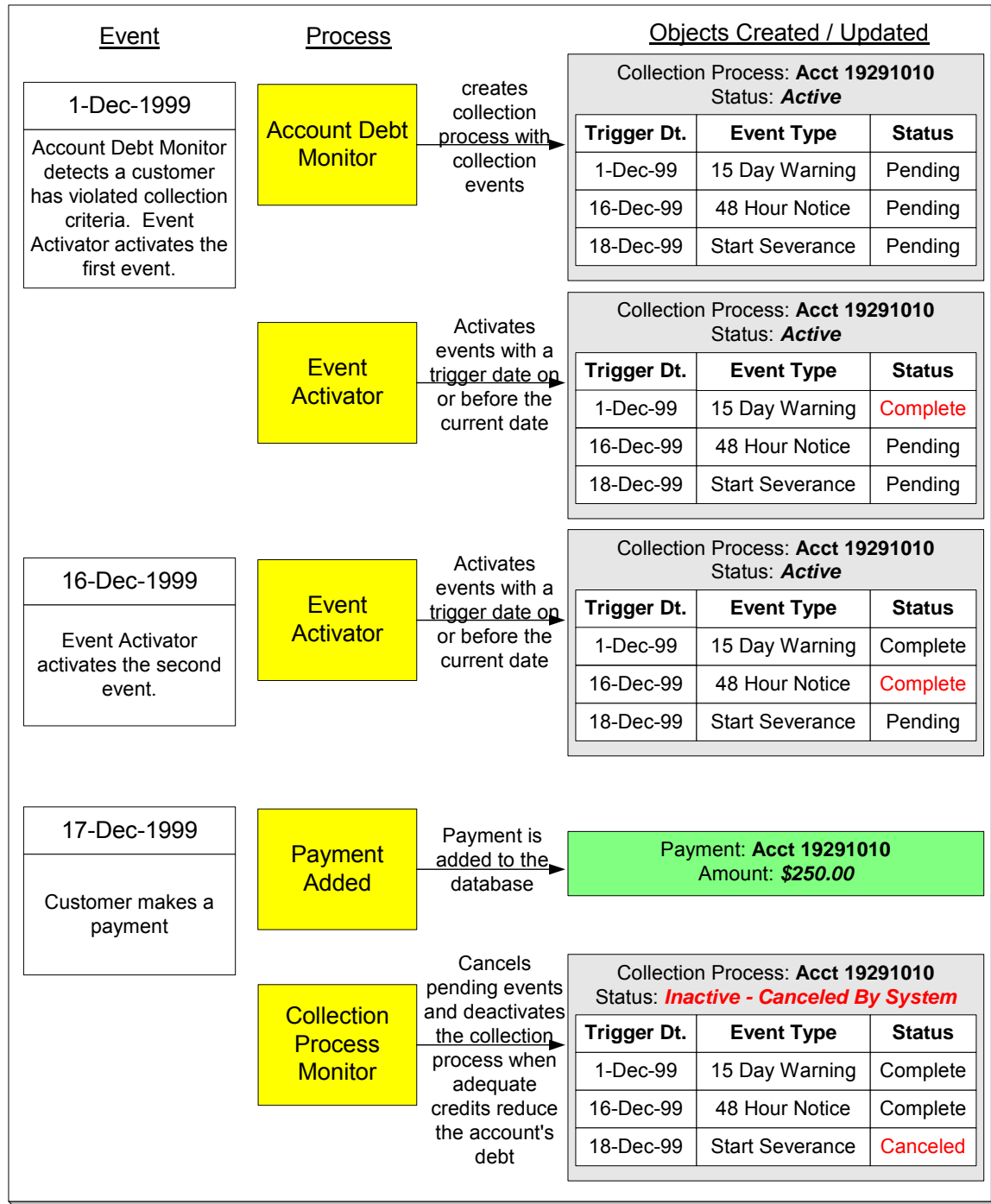
- [Automating Your C&C Activities](#)
- [The Lifecycle Of A Collection Process And Its Events](#)
- [The Collection Process Background Processes](#)
- [The Lifecycle Of A Severance Process And Its Events](#)
- [The Severance Process Background Processes](#)
- [The Lifecycle Of A Write Off Process And Its Events](#)
- [The Write-Off Process Background Processes](#)

Automating Your C&C Activities

Refer to [The Big Picture Of Credit & Collections \(C&C\)](#) to understand how the system monitors overdue debt and initiates the appropriate collection, severance and write-off events when a customer exceeds your debt tolerances.

The Lifecycle Of A Collection Process And Its Events

The following diagram illustrates the lifecycle of a typical collection process and its collection events:



The above diagram is referenced in the following topics.

Contents

- [What Is A Collection Process?](#)
- [How Are Collection Processes Created?](#)
- [How Are Collection Events Completed?](#)
- [How Are Collection Processes \(and their Events\) Canceled?](#)
- [What Happens If A Payment Is Not Received?](#)

What Is A Collection Process?

A collection process is a series of events (e.g., letters, To Do entries) meant to encourage an account to pay its delinquent debt. Linked to the collection process are the specific service agreements that contributed to the delinquent debt.

How Are Collection Processes Created?

A collection process and its events may be created as follows:

- The Account Debt Monitor is a background process that creates a collection process when an account violates your collection criteria. The collection process has one or more collection event(s). The collection process template associated with the collection process controls the number and type of events.
- A user may create an ad hoc collection process at will. The collection process template controls the number and type of events linked to the collection process.

In the above diagram, the event on 1-Dec-99 shows the creation of a collection process with three collection 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 [How Does The Account Debt Monitor Work?](#) to understand how the system monitors overdue debt and initiates the appropriate collection process when a customer exceeds your debt tolerances. Refer to [The Big Picture Of Collection Events](#) for more information about a collection process' events.

When you enable the Control Central alert algorithm, [C1-COLLPR-AC](#), an alert displays when an account has an active collection process. This algorithm is plugged-in on the [installation record](#).

How Are Collection Events Completed?

The event activator runs periodically (at least daily). It looks for collection events with a trigger date on or before the current date. For each such event, the system executes its activity and then completes it.

In the above diagram, the event activator activates pending events on 1-Dec-99 and 16-Dec-99. Notice that the events are moved to the **Complete** state.

Refer to [The Big Picture Of Collection Events](#) for more information about the various types of collection events and what happens when they are completed.

How Are Collection Processes (and their Events) Canceled?

Refer to [How Are Collection Processes Cancelled](#) for a description of how and when collection processes are automatically cancelled by the system.

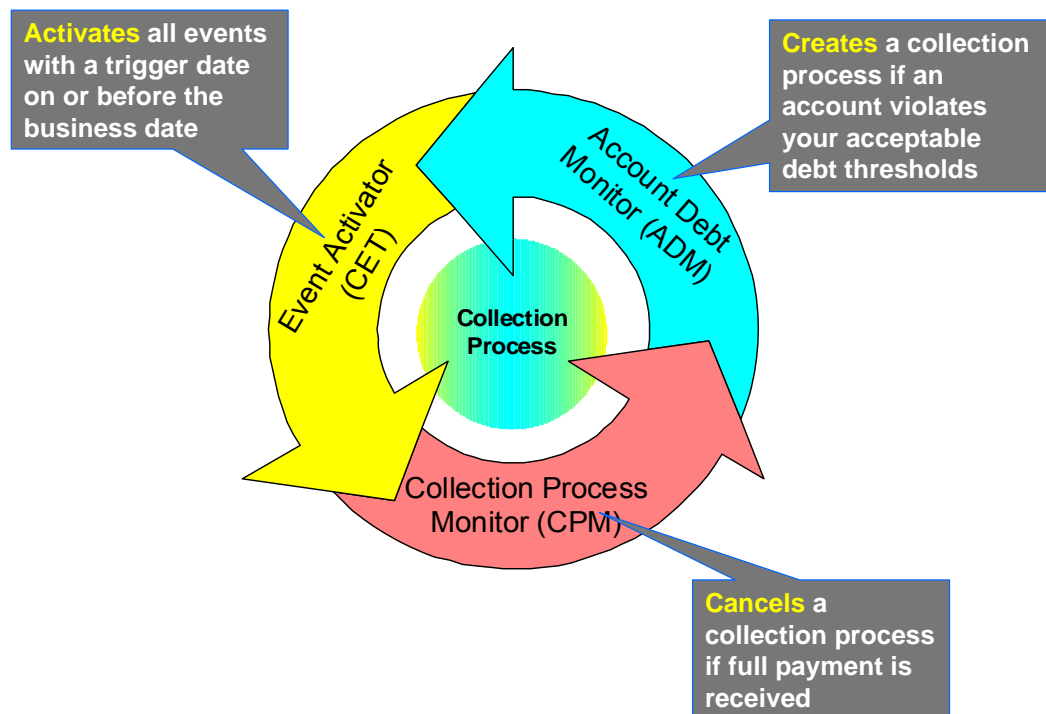
Besides the automated cancellation process, a user may cancel a collection process at will. Refer to [How To Cancel A Collection Process](#) for more information.

What Happens If A Payment Is Not Received?

If a payment (or some other type of credit) is not received by the trigger date of the last event in the process, a severance process is initiated for one or more service agreements. Refer to [The Last Collection Event Should Sever Service Agreement\(s\)](#) for more information.

The Collection Process Background Processes

There are 3 background processes that automate the creation, activation, and cancellation of collection processes:



Contents

- [Account Debt Monitor](#)
- [Collection Event Activator](#)
- [Collection Process Monitor](#)

Account Debt Monitor

The Account Debt Monitor (ADM) creates a collection process when an account violates your collection criteria. The collection process has one or more collection event(s). The number and type of events is defined on the collection process's template.

Heads up. There are actually two background processes involved with the monitoring of overdue debt. These processes are referred to using the batch process ID's of **ADM** and **ADM2**.

In the above [diagram](#), the ADM creates a collection process (with 3 events) on 1-Dec-99. 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 [How Does The Account Debt Monitor Work](#) for more information.

Collection Event Activator

The collection event activator runs periodically (at least daily). It looks for collection events with a trigger date on or before the current date. For each such event, the system executes its activity and then completes it.

Note. This batch process is referred to using the batch ID **CET**.

In the above [diagram](#), the collection event activator activates pending events on 1-Dec-99 and 16-Dec-99. Notice that the events are moved to the **Complete** state after they are activated.

Collection Process Monitor

The Collection Process Monitor runs periodically (at least daily). This process reviews a collection process whenever one of its service agreements' debt is reduced.

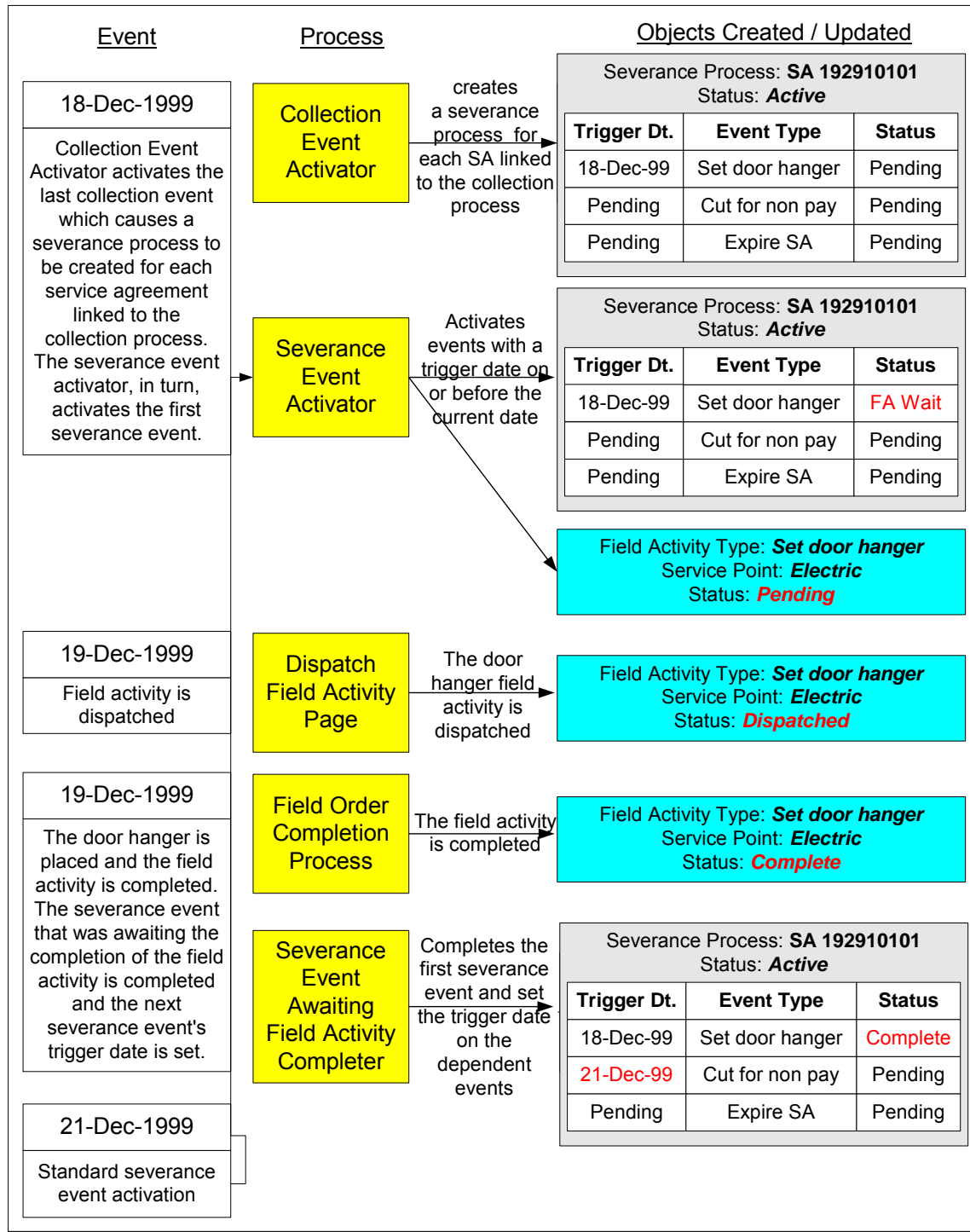
Note. This batch process is referred to using the batch ID **CPM**.

At review time, the Collection Process Monitor determines if it can cancel the collection process (because the customer's debt is no longer in a bad state). Refer to [How Does The Collection Process Monitor Work](#) for details describing how this process works.

In the above [diagram](#), the collection process monitor cancels the process on 17-Dec-99. Notice that the events are moved to the **Cancelled** state and the process becomes **Inactive**.

The Lifecycle Of A Severance Process And Its Events

The following diagram illustrates the first part of the lifecycle of a typical severance process and its events:



The above diagram is referenced in the following topics.

Contents

- [What Is A Severance Process?](#)
- [How Are Severance Processes Created?](#)
- [How Are Severance Events Completed?](#)

How Are Severance Processes (and their Events) Canceled? What Happens If The Customer Does Not Pay?

What Is A Severance Process?

A severance process is a series of events (e.g., letters, To Do entries, field activities) that lead to the severance of a service agreement. A separate severance process is required for each service agreement to be severed.

How Are Severance Processes Created?

Severance processes may be created as follows:

- The Collection Event Activator creates a severance process when it completes a “start severance process” collection event. The severance process has one or more severance event(s). The severance process template associated with the severance process controls the number and type of events.
- A user may create an ad hoc severance process at will. The severance process template defined when the severance process is created controls the number and type of events linked to the severance process.

Manual Creation. A user can create a severance process for an account that does not qualify to be on severance according to the cancel criteria algorithm. For example, perhaps your cancel criteria algorithm cancels a severance process when the account's debt falls below a threshold amount. A user can create a severance process for an account whose debt is already below this threshold. Because cancellation is real time, there is no action that will cause this severance process to be canceled. When a manual severance process is created, the system executes the appropriate cancellation criteria algorithm. If the algorithm indicates that the system would have canceled this severance process, a warning is issued. Refer to [How Are Severance Processes Canceled](#) for more information about cancellation algorithms.

In the above diagram, the event on 18-Dec-99 shows the creation of a severance process with three collection events. Notice that only the first event has a trigger date, this is because the later events' trigger date can only be determined after the first event is completed. Also, 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 [How Are Collection Events Completed?](#) for more information about the completion of collection events.

When you enable the Control Central Alert installation algorithm, [C1-SEVPR-ACT](#), an alert displays when an account has an active severance process. This algorithm is plugged-in on the [installation record](#).

How Are Severance Events Completed?

The Severance Event Activator (note: this batch process is called **SET**) runs periodically (at least daily). It looks for severance events with a trigger date on or before the current date. For each such event, the system executes its activity. The type of severance event dictates what happens to the severance event's state. Refer to [Field Events Versus Office Events](#) for the details.

In the above diagram, the event activator activates a pending event on 18-Dec-99. Take note of the state of the severance event (it becomes ***Awaiting Field Activity Completion***).

Refer to [The Big Picture Of Severance Events](#) for more information about the various types of severance events and what happens when they are completed.

How Are Severance Processes (and their Events) Canceled?

Refer to [How Are Severance Processes Cancelled](#) for a description of how the system automatically cancels a severance process.

Real time cancellation. Unlike collection processes, the system cancels severance processes real time (i.e., there is no equivalent of the collection process monitor for severance processes). Why are severance processes canceled real time? Because a severance process may have events that create field activities to sever service. These events need to be canceled the moment the FT is frozen, we can't wait until a background process runs. This means that if a customer pays in person for a service agreement that is pending severance, the system will cancel the process and its field activities (if any) the moment the payment is entered. The logic to cancel the field activities is in an algorithm plugged in on the [installation record](#).

Refer to [Setting Up Severance Process Templates](#) for more information about severance process templates. Refer to [Designing Your Reconnection Procedures](#) for more information about how severance processes are used to reconnect services.

Besides the automated cancellation process, a user may cancel a severance process at will.

What Happens If The Customer Does Not Pay?

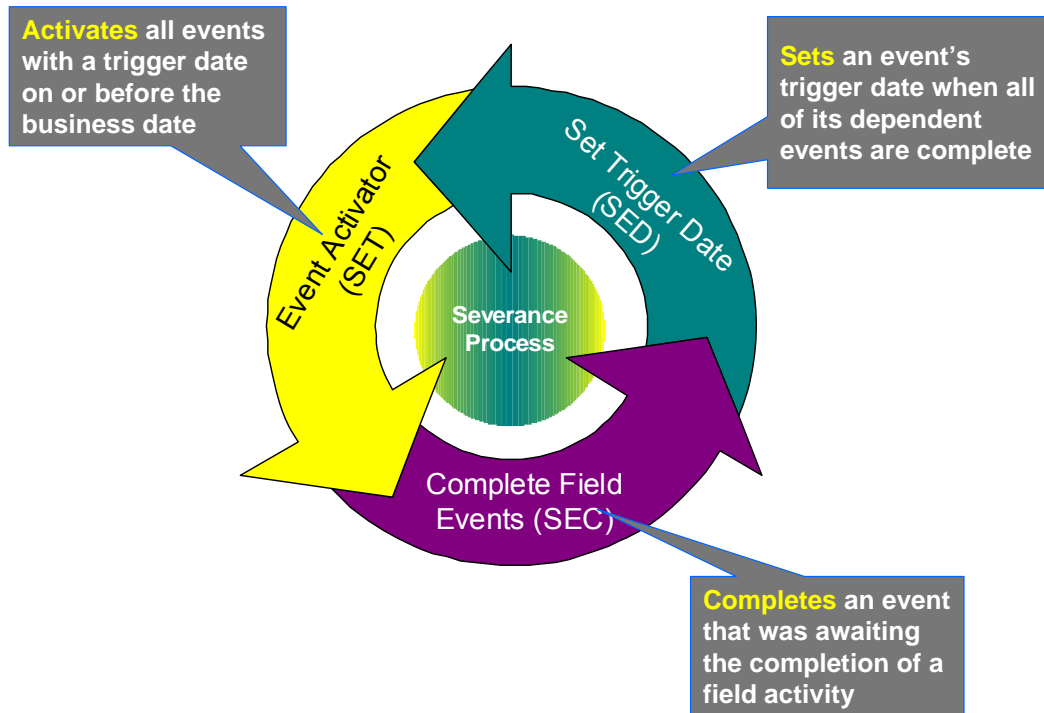
If a payment (or some other type of credit) is not received by the trigger date of the last event in the severance process, the service agreement is stopped (as long as you've setup your severance process to include an ***Expire Service Agreement*** event as the last event). The expiration of the last service agreement will, in turn, trigger a final bill. If the account doesn't pay the final bill, a write-off process will be started.

Final bills. The account will be setup for final billing when the LAST service agreement linked to the account is stopped. Refer to [Finalizing Pending Stops](#) for more information about how the system changes the account's bill cycle when the last service agreement is stopped.

Refer to [The Big Picture Of Write Off Processing](#) for more information about the write off process.

The Severance Process Background Processes

The Collection Event Activator creates a severance process when it completes a “start severance process” collection event. The severance process has one or more severance event(s). The severance process template associated with the severance process controls the number and type of events. There are 3 background processes that automate the activation and cancellation of severance events:



Contents

Severance Event Activator
 Completing Severance Events That Are Dependent On The Completion Of A Field Activity
 Set Trigger Date

Severance Event Activator

The Severance Event Activator runs periodically (at least daily). It looks for severance events with a trigger date on or before the current date. For each such event, the system executes its activity. The type of severance event dictates what happens to the severance event's state.

Note. This batch process is referred to using the batch ID **SET**.

In the above [diagram](#), the event activator activates a pending event on 18-Dec-99. Take note of the state of the severance event (it becomes **Awaiting Field Activity Completion**).

Completing Severance Events That Are Dependent On The Completion Of A Field Activity

The Complete Field Events process runs periodically (at least daily). It looks for severance events that are dependent on the completion of field activities before they can be completed. The system completes each such event when the field activities linked to the event are complete.

Note. This batch process is referred to using the batch ID **SEC**.

In the above [diagram](#), this process completes the event that was **Awaiting Field Activity Completion** on 19-Dec-99. Take note of the state of the severance event (it becomes **Complete**).

Set Trigger Date

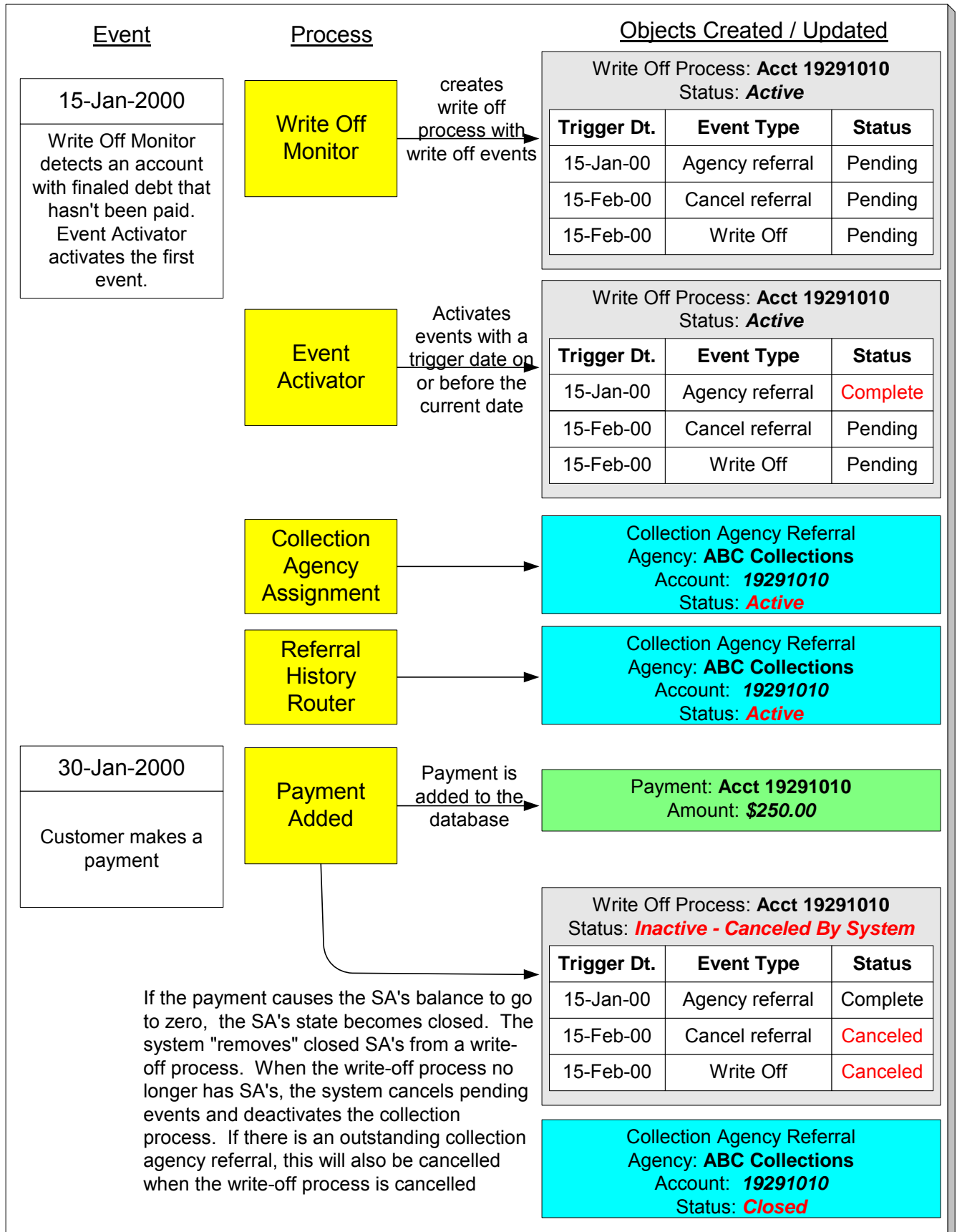
The Set Trigger Date process runs periodically (at least daily). It looks for severance events that are dependent on the completion of other events before their trigger date can be set. The system sets the trigger date of each such event when all of its dependent events are complete.

Note. This batch process is referred to using the batch ID **SED**.

In the above [diagram](#), this process sets the trigger date of the second event on 19-Dec-99.

The Lifecycle Of A Write Off Process And Its Events

The following diagram illustrates the lifecycle of a typical write-off process and its write-off events:



The above diagram is referenced in the following topics.

Contents

- [What Is A Write-off Process?](#)
- [How Are Write-off Processes Created?](#)
- [How Are Write-off Events Completed?](#)
- [How Are Write-off Processes \(and their Events\) Canceled?](#)
- [What Happens If The Debt Is Not Paid?](#)
- [How Is Debt Referred To Collection Agencies](#)

What Is A Write-off Process?

A write-off process is a series of events (e.g., letters, collection agency referrals) meant to encourage an account to pay its delinquent debt. Linked to the write-off process are the specific service agreements that contributed to the delinquent debt.

How Are Write-off Processes Created?

A write-off process and its events may be created as follows:

- The Write Off Monitor is a background process (referred to by the batch ID **WPM**) that creates a write-off process when an account has finaled, unpaid service agreements. The write-off process has one or more write-off event(s). The write-off process template associated with the write-off process controls the number and type of events.
- A user may create an ad hoc write-off process at will. The write-off process template controls the number and type of events linked to the write-off process.

In the above diagram, the event on 15-Jan-00 shows the creation of a write-off process with three write-off 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 [How Does The Write-Off Monitor Work?](#) to understand how the system monitors overdue debt and initiates the appropriate write-off process when a customer has unpaid, finaled service agreements. Refer to [The Big Picture Of Write-off Events](#) for more information about a write-off process' events.

How Are Write-off Events Completed?

The event activator runs periodically (at least daily). It looks for write-off events with a trigger date on or before the current date. For each such event, the system executes its activity and then completes it.

In the above diagram, the event activator activates a pending event on 15-Jan-00. Notice that the events are moved to the **Complete** state.

Refer to [The Big Picture Of Write-off Events](#) for more information about the various types of write-off events and what happens when they are completed.

How Are Write-off Processes (and their Events) Canceled?

When a stopped service agreement's balance is reduced to zero (i.e., it closes), the system checks if there is an ongoing write-off process associated with the service agreement. If so:

- The system “removes” the service agreement from the write-off process.
- When the last service agreement is removed from a write-off process, the system cancels all pending write-off events and deactivates the write-off process.
- If the canceled process has collection agency referrals, the system cancels the referrals.

Real time cancellation. Unlike collection processes, the system cancels write-off processes real time (i.e., there is no equivalent of the collection process monitor for write-off processes). This means that if a customer (or collection agency) pays for a service agreement that is linked to write-off process, the system will cancel the process and its collection agency referrals (if any) the moment the payment is entered.

Refer to [Setting Up Write Off Process Templates](#) for more information about write off process templates.

Besides the automated cancellation process, a user may cancel a write-off process at will.

What Happens If The Debt Is Not Paid?

If a payment (or some other type of credit) is not received by the trigger date of the last event in the process, the debt will be written off (assuming you designed your write off process template to do such).

The last write-off event should cause debt to be written off. Debt will only be written off if the last write-off event is the type that writes off debt. If the write-off process template contains no such event, nothing further will transpire until the next time the write-off monitor runs (when it will create another write-off process for the unpaid, finalized debt).

Refer to [How Is Debt Financially Written-Off?](#) for a description of exactly how debt is written off.

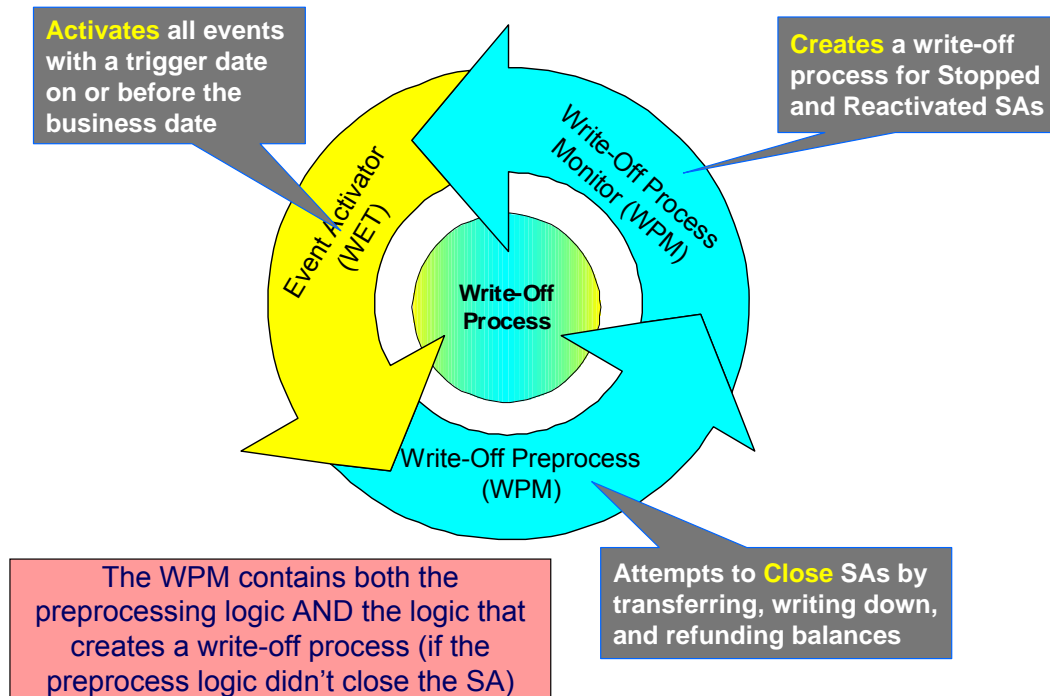
How Is Debt Referred To Collection Agencies

Refer to [How Do Collection Agency Referrals Work?](#) for more information about how the system refers debt to a collection agency.

The Write-Off Process Background Processes

There are 2 background processes that automate the creation, activation, and cancellation of write-off processes:

Note. If necessary, you can write-off debt outside of the automated write-off process by simply transferring bad debt to a write-off service agreement using the Write Off transaction.



Contents

Write-Off Monitor
Write-Off Event Activator

Write-Off Monitor

The Write-Off Monitor is a background process that has two functions:

- It attempts to close Stopped and Reactivated service agreements using several methods as described under [Attempt To Close The SA Before Creating A Write Off Process](#).
- If it is unsuccessful at closing a SA, it creates a write-off process. The write-off process has one or more write-off event(s). The write-off process template associated with the write-off process controls the number and type of events.

Note. This batch process is referred to using the batch ID **WPM**.

In the above [diagram](#), the event on 15-Jan-2000 shows the creation of a write-off process with three write-off 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.

Write-Off Event Activator

The write-off event activator runs periodically (at least daily). It looks for write-off events with a trigger date on or before the current date. For each such event, the system executes its activity and then completes it.

Note. This batch process is referred to using the batch ID **WET**.

In the above [diagram](#), the write-off event activator activates a pending event on 15-Jan-2000. Notice that the event is moved to the **Complete** state.

Collection Process Maintenance

A collection process is a series of events (e.g., letters, To Do entries) meant to encourage an account to pay its delinquent debt. Linked to the collection process are the specific service agreements that contributed to the delinquent debt. The topics in this section describe the pages on which collection process-related information is maintained.

For more information about collection processes, refer to [The Lifecycle Of A Collection Process And Its Events](#).

Contents

- [Collection Process - Main Information](#)
- [Collection Process - SAs \(Service Agreements\)](#)
- [Collection Process - Events](#)
- [How To Perform Common Collection Process Functions](#)

Collection Process - Main Information

The Main page contains core collection process information. Open this page using **Credit & Collection, Collection Process**.

The **Description of Page** section below describes the fields on this page. Refer to [How To Perform Common Collection Process Functions](#) for more instructions describing how to use this transaction.

Description of Page

Collection Process is a concatenation of summary information about this collection process. It is composed of the name of the main customer on the account, the collection status, the reason code for the collection status, the collection class control name, and the creation date of the process.

Collection Process ID is the unique system-generated identifier for this collection process.

Account ID is the system-generated account number. The name of the main customer on the account and its customer class are displayed next to the ID number.

Collection Status defines the state of the collection process. The following values may exist:

Active

The system creates a collection process in the **Active** state. A collection process remains in this state until the last event linked to the process is complete OR the customer pays the collection amount on the service agreements linked to the process OR you want to stop the process for whatever reason.

Inactive

A collection process becomes **Inactive** when the last event linked to the process is completed OR the customer pays the collection amount on all service agreements linked to the process OR you want to stop the process for whatever reason. The **Reason** field tells you how a collection process became inactive.

To completely understand the significance of a collection process' status, refer to [The Lifecycle Of A Collection Process And Its Events](#).

Reason defines how a collection process became **Inactive** or why a collection process is **Active**. The following values may exist:

Canceled by User

This reason should be used when a user manually inactivates a collection process (for whatever reason).

Canceled by System

This reason is populated by the system when it cancels a collection process. The system automatically cancels a collection process when sufficient credits have been posted to the process' service agreements since the process was started.

Events Pending

This reason is populated by the system when a process has pending events.

Completed

This reason is populated by the system when it completes the last event linked to the collection process.

Collection Class Control defines the debt class (i.e., the type of debt) associated with the collection process. This is important as it controls which service agreements can be linked to the collection process (every service agreement is associated with a specific debt class). This field is gray when there are service agreements linked to the collection process.

Multiple collection processes may be linked to an account. It's important to be aware that if an account's service agreements reference multiple debt classes, a collection process must be created for each debt class that you want to collect.

One collection process per debt class. A given debt class for an account may only have one ongoing collection process at any point in time.

Beneath **Collection Class Control** is displayed a message showing why the system generated the collection process. This information is only displayed for collection processes created by the Account Debt Monitor. If you have plugged in a **Collection Process Additional Information** algorithm on the [Installation](#) table, the results of the algorithm, if any, is displayed on a separate line.

The following line displays the minimum value payment to the collection process debt class that will cause cancellation of the collection process.

For more information, refer to [How Does The Account Debt Monitor Work?](#).

Collection Process Template defines the collection 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 Collection Process Templates](#).

Creation Date/Time defines the start date/time of the collection process. This is important as it affects the trigger dates of the process' collection events. This field becomes protected after the collection process exists on the database. If you need to change the trigger date on the events you can navigate to the **Events** tab and change the dates manually. Alternatively, you can perform the following "trick":

- Change the **Collection Process Template** to something (anything). Doing this causes the **Create Date/Time** to become unprotected.
- Specify the date of the first collection event in the newly unprotected **Create Date/Time**.
- Change the **Collection Process Template** back to the desired value. When you change the template, the system removes the events and creates new ones using the **Create Date/Time**.

The collection process cancellation logic will cancel a collection process when its account's debt no longer exceeds a given amount (you define this threshold amount when you set up the system). Rather than compare an account's total debt to the threshold amount, the cancellation logic compares the account's debt that is older than **X** days to the threshold amount. **X** is the date specified in **Collection Amount Base Date**. You can have the system calculate this date for you by entering the age of the arrears in the field embedded in **Calculate Date Using ____ Days In Arrears**.

For more information, refer to [How Are Collection Processes \(and their Events\) Canceled?](#)

Enter any **Comments** about the collection process.

The [tree](#) at the bottom of the page shows a variety of information about the **account** and the collection process including:

- The service agreements linked to the collection process.
- The events linked to the collection process.
- The amount of debt currently associated with the account's debt classes.
- Whether a pay plan exists for the account, or existed at the time the collection process as created.
- The balance for any service agreements linked to the collection process in Inactive status but that have credit balances contributing to the overall debt class arrears for the account.
- A summary of other active collection activities (i.e., collection, severance and write-off processes) associated with the **account**.

Collection Process - SAs (Service Agreements)

The SA page contains the service agreements whose debt is being collected by the collection process. Open this page using **Credit & Collection, Collection Process, SAs**.

The **Description of Page** section below describes the fields on this page. Refer to [How To Perform Common Collection Process Functions](#) for more instructions describing how to use this transaction.

Description of Page

The top portion of every page in this page contains the following information:

Collection Process is a concatenation of summary information about this collection process. It is composed of the name of the main customer on the account, the collection status, the reason code for the collection status, the collection class control name, and the creation date of the process.

ID is the unique system-generated identifier for this collection process.

The **Collection SAs** scroll contains the service agreement being collected under this collection process. To modify a service agreement linked to the collection process, simply move to a field and change its value. To add a new service agreement, press the + button to insert a row, then fill in the information for each field. The following fields display:

Service Agreement ID identifies the service agreement being collected. Information about the service agreement is displayed beneath the SA ID.

Status defines the state of the service agreement in respect of the collection process. The following values may exist:

Active

An **Active** service agreement is one currently being collected by the collection process. The service agreement will remain in this state until the last event linked to the process is complete OR the customer pays the collection amount on the service agreement OR you want to remove the service agreement from the collection process for whatever reason.

Inactive

A service agreement becomes **Inactive** within the collection process when the last event linked to the process is completed OR the customer pays the collection amount on the service agreement OR you want to remove the service agreement from the collection process for whatever reason.

The **Collection Amount** is the amount of debt that initially triggered the collection process. This will be zero for collection processes that were created manually.

The **Amount Still Owing** is the amount of debt older than the **Collection Amount Base Date** defined on the Main page.

The **Credit Amount** is displayed if the Amount Still Owing is less than zero.

The **SA Balances** area displays how much is owed for the service agreement. The **Days Old** and **Arrears Amount** information shows how old the current balance is. This information will be blank if the current balance is a credit.

Note. Information about debt balances by age is not applicable to [open-item accounts](#).

Current Balance indicates the amount the customer currently owes for the service. **Payoff Balance** indicates the amount the customer would owe if they wanted to close the service agreement. This value is only displayed when it differs from the current balance.

Refer to [Current Amount versus Payoff Amount](#) for more information.

Collection Process - Events

The events page contains the activities that will be performed to persuade the customer to pay the outstanding debt. Open this page using **Credit & Collection, Collection Process, Events**.

The **Description of Page** section below describes the fields on this page. Refer to [How To Perform Common Collection Process Functions](#) for more instructions describing how to use this transaction.

We strongly recommend that you understand the information described in [How Are Collection Events Completed?](#) before using this page.

The number and type of events associated with a collection process are defined by the process's collection process template (which is defined on the first page). Refer to [Setting Up Collection Process Templates](#) for more information.

Description of Page

The top portion of every page in this page contains the following information:

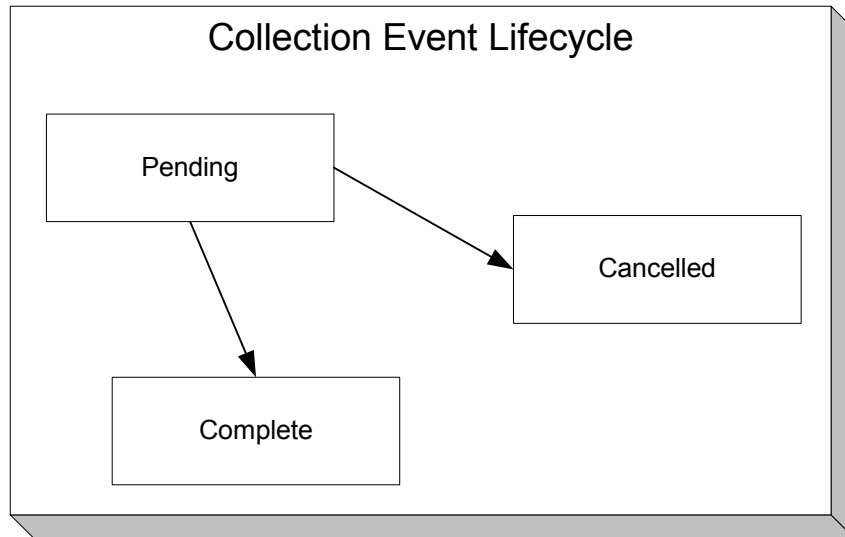
Collection Process is a concatenation of summary information about this collection process. It is composed of the name of the main customer on the account, the collection status, the reason code for the collection status, the collection class control name, and the creation date of the process.

ID is the unique system-generated identifier for this collection process.

The **Collection Events** scroll contains the collection events that comprise the collection process. Only one event is displayed at a time. The inner scroll controls the display of the customer contacts that were created by the system when a particular collection event was triggered.

Event Sequence is the unique identifier of the event.

Collection Event Status defines the state of the event. The following diagram shows the possible lifecycle of a collection event:



Collection events are initially created in the **pending** state.

When the system sees a **pending** event with a **Trigger Date** on or before the current date, the system executes the event's activity and **completes** the event.

You can cancel an individual collection event by setting the status to **cancelled**.

The **Collection Event Type** defines the event's activity (e.g., Email sent, a To Do entry is generated, a letter is sent).

For more information, refer to [Setting Up Collection Process Templates](#).

The **Trigger Date** defines the date when the system completes (i.e., executes) the event.

For more information about a collection event's trigger date, see [Collection Event Trigger Date](#).

If the status of the event is **complete**, the **Completion Date** is displayed with the date on which the system completes the event. If the status of the event is **cancelled**, the **Canceled Date** is displayed with the date on which the event was system canceled.

The **Customer Contact** information contains information when the event template causes a letter to be generated (because letters are created by customer contacts).

How To Perform Common Collection Process Functions

The topics in this section describe how to perform common collection process maintenance functions. Refer to [The Lifecycle Of A Collection Process](#) for more information about collection processes. Refer to [The Big Picture Of Collection Processing](#) for high-level information about collection processing.

Contents

- [How To Create A Collection Process](#)
- [How To Change Collection Events](#)
- [How To Cancel A Collection Process](#)

How To Create A Collection Process

99.9% of all collection processes are created by the [Account Debt Monitor](#) and require no human intervention before they are executed (refer to [How Are Collection Processes Created](#) for information about how the account debt monitor creates collection processes). 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 collection process.
- After the account is populated on Control Central, choose the **Collection Process +** option on the account context menu to transfer to the collection process transaction in add mode for the account.
- After the Collection Process – Main page appears, specify the appropriate **Collection Class Control**. **Collection Class Control** defines the debt class (i.e., the type of debt) associated with the collection process. This is important as it controls which service agreements can be linked to the collection process (every service agreement is associated with a specific debt class). It's important to be aware that if an account's service agreements reference multiple debt classes, a collection process must be created for each debt class that you want to collect. Also be aware that a given debt class for an account may only have one ongoing collection process at any point in time.
- Choose a **Collection Process Template**. **Collection Process Template** defines the collection events in the process. You can override these events on the Events page. For more information, refer to [Setting Up Collection Process Templates](#).
- Use **Collection Amount Base Date** to define the arrears cutoff date that the system uses when it "removes" a SA from a collection process. The system removes a SA from a collection process when its debt older than **X** is less than or equal to zero. **X** is the date specified in **Collection Amount Base Date**. You can have the system calculate this date for you by entering the age of the arrears in the field embedded in **Calculate Date Using ____ Days In Arrears**.
- Navigate to the **SAs** page to define the specific service agreement's whose debt is covered by this collection process. You must define at least one service agreement.
- Save the collection process.

How To Change Collection Events

When a collection process is first created, it has one or more collection 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 collection process is initiated are defined on the collection process's collection process template. The following points describe how to add / change / delete events on a collection process if the defaulted events are not satisfactory.

- Use **Control Central** to choose the account with the collection process whose events need to be changed.
- After the account is populated on Control Central, choose the **Collection Process** option on the account context menu to transfer to the collection process transaction in update mode for the account.
- To add a new event, transfer to the **Events** tab and press the + button in the **Collection Events** scroll to add a new event. 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 **Collection Event Status** of *Pending*.
- Choose the desired **Collection Event Type**.
- Use **Trigger Date** to define the date on which the event should be activated (i.e., completed).
- To delete an existing event, transfer to the **Events** tab and use the scroll arrows to toggle to the desired collection event. When the desired collection event appears, press 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, transfer to the **Events** tab and use the scroll arrows to toggle to the desired collection event. When the desired collection event appears, make the desired changes.
- After all desired changes have been made, save the collection process.

How To Cancel A Collection Process

The [Collection Process Monitor](#) will automatically “remove” service agreements from a collection process when their debt has been reduced sufficiently. When all service agreements are “removed” from a collection process, the collection process is canceled. The following points describe how to manually cancel a collection process.

- Use **Control Central** to choose the account with the collection process to be cancelled.
- After the account is populated on Control Central, choose the **Collection Process** option on the account context menu to transfer to the collection process transaction in update mode for the account. A list of all collection processes associated with the account appears. If only one collection process exists, it is automatically selected for you.
- On the **Main** tab, click Cancel.
- When the warning dialog appears, click OK to continue. The collection process and its pending events are canceled.

Severance Process Maintenance

A severance process is a series of events (e.g., field orders, letters, To Do entries) meant to encourage an account to pay its delinquent debt. Linked to the severance process is the specific service agreement being severed. The topics in this section describe the pages on which severance process-related information is maintained.

For more information about severance processes, refer to [The Lifecycle Of A Severance Process And Its Events](#).

Contents

- [Severance Process - Main Information](#)
- [Severance Process - Severance Events](#)
- [Severance Process - Field Activity/Customer Contact](#)

How To Perform Common Severance Process Functions

Severance Process - Main Information

The Main page contains core severance process information. Open this page using **Credit & Collection, Severance Process**.

The **Description of Page** section below describes the fields on this page. Refer to [How To Perform Common Severance Process Functions](#) for more instructions describing how to use this transaction.

Description of Page

The top portion of every page in this page contains the following information:

Severance Process is a concatenation of summary information about this severance process. It is composed of the name of the main customer on the account, the division and SA type, and severance process template code, the process status, the reason code for the process status and the creation date of the process.

Severance Process ID is the unique system-generated identifier for this severance process.

The remaining fields are maintained on the main severance process page.

Service Agreement ID is the system-generated identifier of the service agreement to which the severance process is linked. Adjacent to the ID number is summary information about the SA.

Severance Status defines the state of the severance process. The following values may exist:

Active

The system creates a severance process in the **Active** state. A severance process remains in this state until the last event linked to the process is complete OR the customer pays the collection amount associated with the severance process OR you want to stop the process for whatever reason. The **Reason** field tells you why a severance process is still in the active state.

Inactive

A severance process becomes **Inactive** when the last event linked to the process is completed OR the customer pays the collection amount on the severance process OR you want to stop the process for whatever reason. The **Reason** field tells you how a severance process became inactive.

To completely understand the significance of a severance process' status, refer to [The Lifecycle Of A Severance Process And Its Events](#).

Reason defines how a severance process became **Inactive** or why a severance process is **Active**. The following values may exist:

Canceled by User

This reason should be used when a user manually inactivates a severance process (for whatever reason).

Canceled by System

This reason is specified by the system when it cancels a severance process. The system automatically cancels a severance process when sufficient credits have been posted to the process' service agreement since the process was started.

Events Pending

This reason is specified by the system when a process has pending events.

Completed

This reason is specified by the system when it completes the last event linked to the severance process.

Severance Process Template defines the severance events in the process. You can override these events on the **Severance Events** page. If you change the template after events exist, the system will remove the existing events and replace them with the new template's events.

For more information, refer to [Setting Up Severance Process Templates](#).

Create Date/Time defines the start date/time of the severance process. This is important as it affects the trigger dates of the process' severance events. This field becomes protected after the severance process exists on the database. If you need to change the trigger date on the events you can navigate to the **Events** tab and change the dates manually. Alternatively, you can perform the following "trick":

- Change the **Severance Process Template** to something (anything). Doing this causes the **Create Date/Time** to become unprotected.
- Specify the date of the first severance event in the newly unprotected **Create Date/Time**.
- Change the **Severance Process Template** back to the desired value. When you change the template, the system removes the events and creates new ones using the **Create Date/Time**.

The **Collection Amount** is the amount of debt that initially triggered the severance process.

The **Amount Still Owning** is the amount of debt older than the **Severance Amount Base Date**.

The severance process cancellation logic will cancel a severance process when its SA's debt no longer exceeds a given amount (you define this threshold amount when you set up the system). Rather than compare a SA's total debt to the threshold amount, the cancellation logic compares the SA's debt that is older than **X** days to the threshold amount. **X** is the date specified in **Severance Amount Base Date**. You can have the system calculate this date for you by entering the age of the arrears in the field embedded in **Calculate Date Using ___ Days In Arrears**.

For more information, refer to [How Are Severance Processes \(and their Events\) Canceled?](#)

Enter any **Comments** about the severance process.

The [tree](#) at the bottom of the page shows a variety of information about the **service agreement's** account and the severance process including:

- The events linked to the severance process.
- The amount of debt associated with the account's debt classes.

- A summary of other active severance activities (i.e., severance, severance and write-off processes) associated with the **service agreement's** account.

Severance Process - Severance Events

The events page contains the activities that will be performed to persuade the customer to pay the outstanding debt. Open this page using **Credit & Collections, Severance Process, Events**.

The **Description of Page** section below describes the fields on this page. Refer to [How To Perform Common Severance Process Functions](#) for more instructions describing how to use this transaction.

We strongly recommend that you understand the information described in [How Are Severance Events Completed?](#) before using this page.

The number and type of events associated with a severance process are defined by the process's severance process template (which is defined on the first page). Refer to [Setting Up Severance Process Templates](#) for more information.

Description of Page

The top portion of every page in this page contains the following information:

Severance Process is a concatenation of summary information about this severance process. It is composed of the name of the main customer on the account, the SA business unit and SA type, set ID and severance process template name, the process status, the reason code for the process status and the creation date of the process.

ID is the unique system-generated identifier for this severance process.

The **Severance Events** scroll contains the severance events that comprise the severance process. The grid at the bottom (when visible) shows the events that must be completed before the severance event is triggered. Only one severance event is displayed at a time.

Event Sequence is the unique identifier of the event.

Severance Event Status defines the state of the event.

For more information, refer to [Severance Event Lifecycle](#).

The **Severance Event Type** defines the event's activity (e.g., a field order is generated, a To Do entry is generated, a letter is sent).

For more information, refer to [Setting Up Severance Event Templates](#).

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 [Severance 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 severance event's trigger date, see [Severance Event Dependencies & Trigger Date](#).

Days After Prev 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 [Severance Event Dependencies & Trigger Date](#).

If the status of the event is **complete**, the **Completion Date** is displayed with the date on which the system completes the event. If the status of the event is **cancelled**, the **Canceled Date** is displayed with the date on which the event was canceled.

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). To add additional sequences press the + button (insert row), and define the sequence of the event on which this event depends.

Severance Process - Field Activity/Customer Contact

The Field Activity/Customer Contact page contains information about the objects created when the system completes the event. Open this page using **Credit & Collection, Severance Process, FA/CC**.

What happens when a severance event is completed (i.e., executed) is dictated by the severance event's template. Refer to [Types Of Severance Events](#) for more information.

Description of Page

The top portion of every page in this page contains the following information:

Severance Process is a concatenation of summary information about this severance process. It is composed of the name of the main customer on the account, the SA business unit and SA type, set ID and severance process template name, the process status, the reason code for the process status and the creation date of the process.

ID is the unique system-generated identifier for this severance process.

The **Severance Events** scroll contains the severance events that comprise the severance process. The grids show the objects created when the system completes each event.

The **Field Activity/SP** grid contains information about the field activities that were created when the event was activated. This information will only be populated with the event's event type causes a field activity to be generated.

The **Customer Contact** area contains information about the customer contacts that were created when the event was activated. This information will only be populated with the event's event type causes a customer contact to be created (note, letters are generated via the creation of a customer contact).

How To Perform Common Severance Process Functions

The topics in this section describe how to perform common severance process maintenance functions. Refer to [The Lifecycle Of A Severance Process](#) for more information about severance processes. Refer to [The Big Picture Of Severance Processing](#) for high level information about severance processing.

Contents

- [How To Create A Severance Process](#)
- [How To Change Severance Events](#)
- [How To Cancel A Severance Process](#)

How To Create A Severance Process

99.9% of all severance processes are created when a "start severance" collection event is activated by the [Collection Event Activator](#) and require no human intervention before they are executed (refer to [How Are Severance Processes Created](#) for information about how the account debt monitor creates severance processes). 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 severance process.
- After the account is populated on Control Central, choose the **Severance Process +** option on the account context menu to transfer to the severance process transaction in add mode for the account. If the account has multiple service agreements, you must select the service agreement to be severed.
- After the Severance Process – Main page appears, specify a **Severance Process Template**. **Severance Process Template** defines the severance events in the process. You can override these events on the Events page. For more information, refer to [Setting Up Severance Process Templates](#).
- Use **Severance Amount Base Date** to define the arrears cutoff date that the system uses when it determines if a severance process can be cancelled. The system cancels a severance process when its SA's debt that is older than **X** is less than or equal to zero. **X** is the date specified in **Severance Amount Base Date**. You can have the system calculate this date for you by entering the age of the arrears in the field embedded in **Calculate Date Using ___ Days In Arrears**.
- Save the severance process.

How To Change Severance Events

When a severance process is first created, it has one or more severance 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 severance process is initiated are defined on the severance process's severance process template. The following points describe how to add / change / delete events on a severance process if the defaulted events are not satisfactory.

- Use **Control Central** to choose the account with the severance process whose events need to be changed.
- After the account is populated on Control Central, choose the **Severance Process** option on the account context menu to transfer to the severance process transaction in update mode for the account.
- To add a new event, transfer to the **Events** tab and press the + button in the **Severance Events** scroll to add a new event. 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 **Severance Event Status** of *Pending*.
 - Choose the desired **Severance 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 a) specify the sequences of the dependent events in **Event Dependencies** and b) specify the number of days after the completion of the last dependent event when the new event should be activated.
 - 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, transfer to the **Events** tab and use the scroll arrows to toggle to the desired severance event. When the desired severance event appears, press 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, transfer to the **Events** tab and use the scroll arrows to toggle to the desired severance event. When the desired severance event appears, make the desired changes.
- After all desired changes have been made, save the severance process.

How To Cancel A Severance Process

The system will automatically cancel a severance process when its service agreement's debt has been reduced sufficiently. Refer to [How Are Severance Processes and Events Cancelled](#) for more information.

The following points describe how to manually cancel a severance process.

- Use **Control Central** to choose the account with the severance process to be cancelled.
- After the account is populated on Control Central, choose the **Severance Process** option on the account context menu to transfer to the severance process transaction in update mode for the account. A list of all severance processes associated with the account appears. If only one severance process exists, it is automatically selected for you.
- On the **Main** tab, click Cancel.
- When the warning dialog appears, click OK to continue. The severance process and its events are canceled.

Write-off Process Maintenance

A write-off process is a series of events (e.g., letters, collection agency referrals) meant to encourage an account to pay its unpaid, finaled debt. Linked to the write-off process are the specific service agreements that contributed to the delinquent debt. The topics in this section describe the pages on which write-off process-related information is maintained.

For more information about write-off processes, refer to [The Lifecycle Of A Write Off Process And Its Events](#).

Contents

- [Write-off Process - Main Information](#)
- [Write-off Process - SAs \(Service Agreements\)](#)
- [Write-off Process - Events](#)
- [How To Perform Common Write-off Process Functions](#)

Write-off Process - Main Information

The Main page contains core write-off process information. Open this page using **Credit & Collections, Write Off Process**.

The **Description of Page** section below describes the fields on this page. Refer to [How To Perform Common Write-Off Process Functions](#) for more instructions describing how to use this transaction.

Description of Page

The top portion of every page contains the following information:

Write Off Process is a concatenation of summary information about this write-off process. It is composed of the name of the main customer on the account, the write-off status, the reason code for the write-off status, the write-off class control name, and the creation date of the process.

Write Off Process ID is the unique system-generated identifier for this write-off process.

The remaining fields are maintained on the main write-off process page.

Account ID is the system-generated account number. The name of the main customer on the account and its customer class are displayed adjacent.

Write-off Status defines the state of the write-off process. The following values may exist:

Active The system creates a write-off process in the **Active** state. A write-off process remains in this state until the last event linked to the process is complete OR the finaled debt is paid off OR you want to stop the process for whatever reason.

Inactive A write-off process becomes **Inactive** when the last event linked to the process is completed OR the finaled debt is paid off OR you want to stop the process for whatever reason. The **Reason** field tells you how a write-off process became inactive.

To completely understand the significance of a write-off process' status, refer to [The Lifecycle Of A Write Off Process And Its Events](#).

Reason defines how a write-off process became *Inactive* or why a write-off process is *Active*. The following values may exist:

<i>Canceled by User</i>	This reason should be used when a user manually inactivates a write-off process (for whatever reason).
<i>Canceled by System</i>	This reason is populated by the system when it cancels a write-off process. The system automatically cancels a write-off process when all finalized debt associated with the process' service agreements is paid.
<i>Events Pending</i>	This reason is populated by the system when a process has pending events.
<i>Completed</i>	This reason is populated by the system when it completes the last event linked to the write-off process.

Write Off Control defines the write-off debt class (i.e., the type of debt) associated with the write-off process. This is important as it controls which service agreements can be linked to the write-off process (every service agreement is associated with a specific write-off debt class). This field is gray when there are service agreements linked to the write-off process.

Multiple write-off processes may be linked to an account. It's important to be aware that if an account's service agreements reference multiple write-off debt classes, a write-off process is created for each write-off debt class that you want to write-off.

Multiple write-off processes per write-off debt class. A given write-off debt class for an account may have multiple ongoing write-off processes at any point in time. However, a given service agreement may only be on one active write-off process at any point in time.

Write Off Process Template defines the write-off 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 Write Off Process Templates](#).

Create Date/Time defines the start date/time of the write-off process. This is important as it affects the trigger dates of the process' write-off events. This field becomes protected after the write-off process exists on the database. If you need to change the trigger date on the events you can navigate to the **Events** tab and change the dates manually. Alternatively, you can perform the following "trick":

- Change the Write-off Process Template to something (anything). Doing this causes the **Create Date/Time** to become unprotected.

- Specify the date of the first write-off event in the newly unprotected **Create Date/Time**.
- Change the **Write-off Process Template** back to the desired value. When you change the template, the system removes the events and creates new ones using the **Create Date/Time**.

Enter any **Comments** about the write-off process.

The [tree](#) at the bottom of the page shows a variety of information about the **account** and the write-off process including:

- The service agreements linked to the write-off process.
- The events linked to the write-off process.
- The amount of debt associated with the account's debt classes.
- A summary of other active collection activities (i.e., collection, severance and write-off processes) associated with the **account**.

Write-off Process - SAs (Service Agreements)

The SA page contains the service agreements whose unpaid, finaled debt is being managed by the write-off process. Open this page using **Credit & Collections, Write Off Process, SAs**.

The **Description of Page** section below describes the fields on this page. Refer to [How To Perform Common Write-Off Process Functions](#) for more instructions describing how to use this transaction.

Description of Page

The top portion of every page in this page contains the following information:

Write Off Process is a concatenation of summary information about this write-off process. It is composed of the name of the main customer on the account, the write-off status, the reason code for the write-off status, the write-off control name, and the creation date of the process.

ID is the unique system-generated identifier for this write-off process.

To modify a service agreement linked to the write-off process, simply move to a field and change its value. To add a new service agreement, press + to insert a row, then fill in the information for each field. The following fields display:

Service Agreement identifies the service agreement being collected. Information about the service agreement is displayed adjacent.

Write Off SA Status defines the state of the service agreement in respect of the write-off process. The following values may exist:

Active

An **Active** service agreement is one currently being collected by the write-off process. The service agreement will remain in this state until the last event linked to the process is complete OR the service agreement is closed (due to payment in full) OR you want to remove the service agreement from the write-off process for whatever reason.

Inactive

A service agreement becomes **Inactive** when the last event linked to the process is completed OR the service agreement is closed (due to payment in full) OR you want to remove the service agreement from the write-off process for whatever reason

The **Write Off Amount** is the amount of debt that initially triggered the write-off process.

The **SA Balances** area displays how much is owed for the service agreement. The **Days Old** and **Amount** information shows how old the current balance is. This information will be blank if the current balance is a credit.

Current Balance indicates the amount the customer currently owes for the service. **Payoff Balance** indicates the amount the customer would owe if they wanted to close the service agreement. This value is only displayed when it differs from the current balance.

Refer to [Current Amount versus Payoff Amount](#) for more information.

Write-off Process - Events

The events page contains the activities that will be performed to persuade the customer to pay the outstanding debt. Open this page using **Credit & Collections, Write Off Process, Events**.

The **Description of Page** section below describes the fields on this page. Refer to [How To Perform Common Write-Off Process Functions](#) for more instructions describing how to use this transaction.

We strongly recommend that you understand the information described in [How Are Write-off Events Completed?](#) before using this page.

The number and type of events associated with a write-off process are defined by the process's write-off process template (which is defined on the first page). Refer to [Setting Up Write Off Process Templates](#) for more information.

Description of Page

The top portion of every page in this page contains the following information:

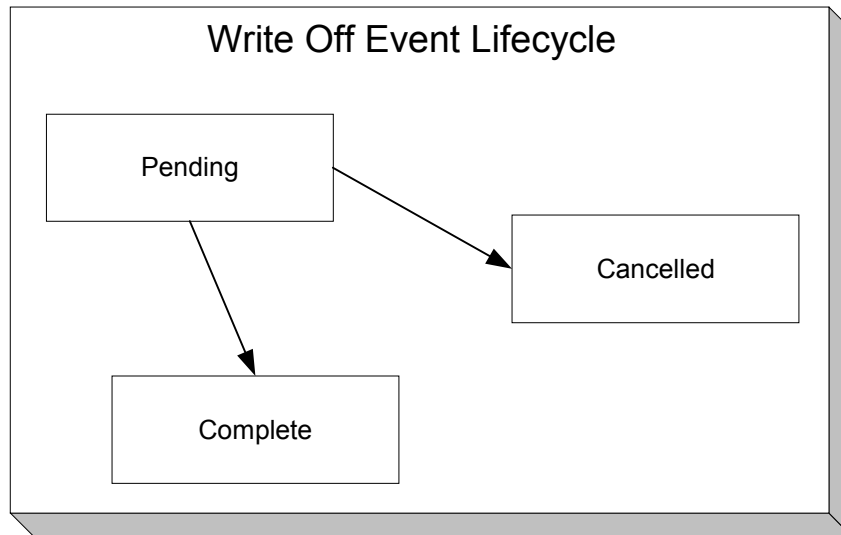
Write Off Process is a concatenation of summary information about this write-off process. It is composed of the name of the main customer on the account, the write-off status, the reason code for the write-off status, the write-off class control name, and the creation date of the process.

ID is the unique system-generated identifier for this write-off process.

The **Write Off Events** scroll contains individual write-off events that comprise the write-off process. Only one event is displayed at a time. The grid contains the customer contacts that were created by the system when a particular write-off event was triggered (if any).

Event Sequence is the unique identifier of the event.

Write Off Event Status defines the state of the event. The following diagram shows the possible lifecycle of a write-off event:



Write-off events are initially created in the **pending** state.

When the system sees a **pending** event with a **Trigger Date** on or before the current date, the system executes the event's activity and **completes** the event.

You can cancel an individual write off event by setting the status to **cancelled**.

The **Write Off Event Type** defines the event's activity (e.g., the related debt is referred to a collection agency, a To Do entry is generated, a letter is sent).

For more information refer to [Setting Up Write Off Process Templates](#).

The **Trigger Date** defines the date when the system completes (i.e., executes) the event.

For more information about a write-off event's trigger date, see [Write-off Event Trigger Date](#).

If the status of the event is **complete**, the **Completion Date** is displayed with the date on which the system completed the event. If the status of the event is **cancelled**, the **Canceled Date** is displayed with the date on which the event was cancelled.

The **Customer Contact** information contains information when the event template causes a letter to be generated (because letters are created by customer contacts).

How To Perform Common Write-off Process Functions

The topics in this section describe how to perform common write-off process maintenance functions. Refer to [The Lifecycle Of A Write-off Process](#) for more information about write-off processes. Refer to [The Big Picture Of Write-off Processing](#) for high level information about write-off processing.

Contents

[How To Create A Write-off Process](#)

[How To Change Write Off Events](#)
[How To Cancel A Write Off Process](#)

How To Create A Write-off Process

99.9% of all write-off processes are created by the [Write-Off Monitor](#) and require no human intervention before they are executed (refer to [How Are Write-off Processes Created](#) for information about how the account debt monitor creates write-off processes). 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 write-off process.
- After the account is populated on Control Central, choose the **Write-off Process +** option on the account context menu to transfer to the write-off process transaction in add mode for the account.
- After the Write-off Process – Main page appears, specify the appropriate **Write Off Control**. **Write Off Control** defines the debt class (i.e., the type of debt) associated with the write-off process. This is important as it controls which service agreements can be linked to the write-off process (every service agreement is associated with a specific debt class). It's important to be aware that if an account's service agreements reference multiple write-off debt classes, a write-off process must be created for each write-off debt class that you want to collect.
- Choose a **Write Off Process Template**. **Write Off Process Template** defines the write-off events in the process. You can override these events on the Events page. For more information, refer to [Setting Up Write-off Process Templates](#).
- Navigate to the **SAs** page to define the specific service agreement's whose debt is covered by this write-off process. You must define at least one service agreement.
- Save the write-off process.

How To Change Write Off Events

When a write-off process is first created, it has one or more write-off events. The events are the activities that will be performed in the course of the write-off process.

The number and type of events that are created when a write-off process is initiated are defined on the write-off process's write-off process template. The following points describe how to add / change / delete events on a write-off process if the defaulted events are not satisfactory.

- Use **Control Central** to choose the account with the write-off process whose events need to be changed.
- After the account is populated on Control Central, choose the **Write Off Process** option on the account context menu to transfer to the write-off process transaction in update mode for the account.
- To add a new event, transfer to the **Events** tab and press the + button in the **Write Off Events** scroll to add a new event. 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 **Write Off Event Status** of *Pending*.
 - Choose the desired **Write Off Type Code**.

- Use **Trigger Date** to define the date on which the event should be activated (i.e., completed).
- To delete an existing event, transfer to the **Events** tab and use the scroll arrows to toggle to the desired write-off event. When the desired write-off event appears, press 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, transfer to the **Events** tab and use the scroll arrows to toggle to the desired write-off event. When the desired write-off event appears, make the desired changes.
- After all desired changes have been made, save the write-off process.

How To Cancel A Write Off Process

The system will automatically “remove” service agreements from a write-off process when the close (i.e., when their balance becomes zero). When all service agreements are “removed” from a write-off process, the write-off process is cancelled. Refer to [How Are Write off Processes and Events Cancelled](#) for more information.

The following points describe how to manually cancel a write-off process.

- Use **Control Central** to choose the account with the write-off process to be cancelled.
- After the account is populated on Control Central, choose the **Write Off Process** option on the account context menu to transfer to the write-off process transaction in update mode for the account. A list of all write-off processes associated with the account appears. If only one write-off process exists, it is automatically selected for you.
- On the **Main** tab, click Cancel.
- When the warning dialog appears, click OK to continue. The write off process and its pending events are canceled.

Writing Off Uncollectable Debt

The Write-off pages described in this section describe the transaction used to manually transfer bad debt to a write-off service agreement.

Contents

[Write Off - Main](#)
[How To Manually Write-Off Debt](#)
[Write Off - Write Off SAs](#)

Write Off - Main

Most write-offs happen behind-the-scenes as part of the automated write-off processing. If you need to perform a partial write-off or cannot wait for the automated process, open **Credit & Collection, Write Off**.

Refer to [The Big Picture of Write Off Processing](#) for background information about the write-off process.

Description of Page

Account contains the name of the main customer on the account. The **Account ID** is the account whose debt is being written off.

The **Candidate SAs** scroll contains one row for each of the account's service agreements that is eligible for write-off. The system displays the respective age and amount of each SA's debt. The accumulated total of all such debt appears in **Write Off Amount**. The checkbox adjacent to this field is used to confirm that you want to write-off the adjacent debt.

When you click the **Create** button, the system writes off the selected debt by executing the write-off algorithm that's plugged in on the account's [customer class](#). Refer to [The Ramifications of Write Offs in the General Ledger](#) for an explanation of how the [base-package algorithm](#) works.

How To Manually Write-Off Debt

To manually write-off debt:

- Select the **Account** in question on [Write Off - Main](#).
- Select the debt to be written off for each **Candidate SA**. You do this by turning on the checkbox adjacent to **Write Off Amount**. You can also vary the amount of the **Write Off Amount**.
- Click the **Create** button.

Write Off - Write Off SAs

The Write Off SAs page shows all write-off service agreements linked to the account along with their respective current and payoff balance. Use **Credit & Collection, Write Off, Write Off SAs** to open this page.

Description of Page

Account contains the name of the main customer on the account. The **Account ID** is the account associated with the WO SAs displayed below.

The grid contains an entry for every write-off service agreement linked to the account. The following information is displayed for each write-off service agreement:

- The service agreement's **SA Information** is displayed.
- **Current Balance** contains the WO SA's current balance (i.e., the amount written off).
- **Payoff Amount** contains the WO SA's payoff balance (i.e., the amount written off).

Write off SA. There is a field on SA type called [Special Role Flag](#). Only those SA types with a role of **Write Off** are displayed.

Collection Referral

The Collection Referrals page contains information about an account's debt that has been referred to a collection agency.

Refer to [How Do Collection Agency Referrals Work?](#) for more information about how the system creates and maintains collection agency referrals as part of the automated write off processing. In theory, you need only access this page if you need to override the automated processing.

Open **Credit & Collections, Collection Agency Referral** to maintain this information.

Description of Page

The **Agency Referrals** scroll contains one entry for every collection agency referral associated with the account. The following information is defined for each referral.

Use **Collection Agency** to define the agency to which the referral is being sent.

Start Date is the date on which the referral was initially created.

Referral Status defines if the referral is **Active** or **Closed**.

Use **Comments** to describe anything unusual about the referral.

The grid contains the history of interactions with the collection agency. Each time an account's debt is referred to a collection agency, the system creates a referral history record.

You can communicate changes about the referral by inserting a new row in the collection. For example,

- If you need to change the referral amount, insert a row and indicate the **Creation Date** and a **Referral History Reason** of **Change Referral**.
- If you need to cancel the referral, insert a row and indicate a **Referral History Reason** of **Referral Cancellation**.
- If the customer pays, insert a row and indicate a **Referral History Reason** of **Referral Paid**.
- If you need to add a new referral, insert a row and indicate a **Referral History Reason** of **Initial Referral**.

Collection agency referral records are interfaced to the respective collection agency using the batch process defined on the collection agency control record (refer to [Setting Up Collection Agencies](#)). The **Batch Control** process and the respective **Batch Number** in which the records were interfaced to the collection agency are displayed adjacent.

The **System Generated** switch is on for those collection agency referrals created as part of the automated write-off processing. Refer to [How Do Collection Agency Referrals Work?](#) for more information.

Payment Arrangements and Pay Plans

Payment arrangements and pay plans are very similar concepts in that they are both used to payoff debt. This section describes each method.

Note. To distinguish between payment arrangements and pay plans, it is helpful to understand some terminology: 1) an agreement with a customer to pay off their debt as part of their future bills (sometimes referred to as "current bill plus") is a **payment arrangement**, 2) an agreement with specifically scheduled payments is called a **pay plan**.

Contents

[Setting Up Payment Arrangements](#)
[Setting Up Pay Plans](#)

Setting Up Payment Arrangements

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. Refer to [The Big Picture Of Payment Arrangements](#) for information about the financial ramifications of payment arrangements and for a description of how the system monitors payment arrangements.

The topics in this section describe how to set up a payment arrangement and how to break a payment arrangement (if a customer doesn't fulfill their obligations).

Contents

[Pay Arrangement - Main](#)
[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](#)
[Pay Arrangement - History](#)

Pay Arrangement - Main

This page allows you to set up a payment arrangement. When you create a payment arrangement, the system performs the following functions:

- It creates a payment arrangement service agreement (PA SA). It sets the installment amount on this service agreement equal to the installment amount specified on this page.
- It transfers delinquent debt from each delinquent service agreement(s) to the new PA SA.
- As it transfers the debt to the PA SA, it creates synchronizing adjustments to reduce the PA SA's current balance by the amount transferred. 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 payoff the debt in installments (and therefore they don't currently owe anything on the PA SA).

Payment Arrangement Algorithm. The system only performs the above steps if the new PA SA's SA Type does not reference a Payment Arrangement algorithm. If your implementation uses a Payment Arrangement algorithm, then the logic performed when a payment arrangement SA is created will depend on this algorithm.

You could do the above functions by adding a new service agreement (using the service agreement page) and creating transfer adjustments (using the adjustment page). However, this is tedious. Rather, open **Credit & Collection, Pay Arrangement** to set up or add additional debt to a payment arrangement.

Description of Page

SA (service agreement) **Info** and **SA ID** are displayed on every page. These values only appear after the payment arrangement service agreement (PA SA) exists on the database. The **ID** is a system assigned random number that stays with the payment arrangement service agreement for life. The **SA Info** is a concatenation of important details about the payment arrangement service agreement and its account.

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 contains the name of the main customer on the account. **Account ID** is the account associated with the PA SA.

The occurrences of **Current** and **Payoff Balance** below **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 **Candidate SAs** scroll contains one row for each of the account's service agreements with aged debt. The system displays the respective age and amount of each SA's debt. All debt that is checked will be transferred to the PA SA when you click **Create**.

Warning! The system automatically indicates that all debt that isn't "new" (i.e., debt that the customer has seen on their bills) should be transferred to the payment arrangement. It does this by checking each parcel of aged debt.

Total is the sum of ALL current balances (not just the checked amounts) from all candidate SAs.

The next two fields are used to populate the arrangement amount (i.e., the amount the customer will be billed on each bill).

- The **Installments** field exists to calculate the **Arrange Amount** (the next field) when a time period has been stipulated. Simply enter the number of installments and tab out. 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.

- Use the **Arrange Amount** field to define the payment arrangement's installment amount if a specific dollar amount has been stipulated. 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.

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.

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 **Create** (or **Change**) causes the system to perform the following:

- It adds / 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 delinquent debt from the delinquent service agreement(s) to the PA SA. The adjustment type used to transfer these funds is defined on the PA SA's SA type.
- 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.

Payment Arrangement Algorithm. The system only performs the above steps if the new PA SA's SA Type does not reference a Payment Arrangement algorithm. If your implementation uses a Payment Arrangement algorithm, then the logic performed when a payment arrangement SA is created will depend on this algorithm.

You can transfer to the **History** page to view the transfer adjustments that were created by the system when it transferred the selected debt to the PA SA.

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 PA SA's SA type). An implementation can use either a Break Payment Arrangement algorithm which the system calls when a payment arrangement is broken, or the Payment Arrangement algorithm which the system calls when a payment arrangement is created, canceled or broken. These two algorithms are mutually exclusive on the SA Type.

Break Payment Arrangement Algorithm. The [base package algorithm](#) expires the PA SA and cancels all related frozen adjustments, returning the debt to the original service agreements. 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.

How To Set Up A Payment Arrangement

To set up a new payment arrangement:

- Select the **Account** in question.
- Select the debt to be transferred to the payment arrangement (by turning on/off the checkboxes adjacent to the **Candidate SA's** amounts).
- Enter the number of **Installments** and the total **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:

- Select the **SA ID** of the existing payment arrangement.
- Select the debt to be transferred to the payment arrangement (by turning on/off the checkboxes adjacent to the **Candidate SA's** amounts).
- Enter the number of **Installments** and the total **Arrange Amount**.
- Click the **Create** 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, i.e., the amount that credit and collections monitors). 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 credit and collections processes will break a payment arrangement behind-the-scenes. Refer to [The Big Picture Of Payment Arrangements](#) for information about how the system monitors payment arrangements.

If you want to manually break a payment arrangement simply select the **SA ID** of the existing payment arrangement and 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

- Select the **SA ID** of the existing payment arrangement and navigate to the **History** tab.
- Click the **Cancel** button to cancel the transfer adjustments that were used to transfer debt to the payment arrangement from the customer's regular service agreements.
- Next, navigate to the payment arrangement SA's (PA SA) financial history page (using the service agreement context menu). You will see that there are a few non-canceled adjustments (these were used to change the PA SA's current balance after debt was transferred to the PA SA). Cancel these adjustments by drilling down on these adjustments and then click the **Cancel** button on the adjustment maintenance transaction.
- Finally, display the PA SA on the service agreement maintenance transaction and cancel the payment arrangement.

Payment Arrangement Algorithm. If the payment arrangement SA's SA Type references a Payment Arrangement algorithm, then the logic performed when the PA SA is canceled will depend on this algorithm.

Pay Arrangement - History

Open **Credit & Collection, Pay Arrangement, History** to view the transfer adjustments that were created by the system when it transferred the selected debt to the PA SA.

Description of Page

SA (service agreement) **Info** and **ID** are displayed on every page in this page. These values only appear after the payment arrangement service agreement (PA SA) exists on the database. The ID is a system-assigned random number that stays with the service agreement for life. The SA Info is a concatenation of important details about the service agreement and its account.

Current Balance 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 contains the name of the main customer on the account. **Account ID** is the account associated with the PA SA.

The occurrences of **Current** and **Payoff Balance** below **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 **Adjustment Details** area contains one row for each transfer adjustment used to transfer delinquent debt to the PA SA. Click the Go To button to transfer to the adjustment page to view the details of an adjustment (or to cancel the adjustment if you want to reinstate the debt on the originating service agreement).

Total Frozen Adj is the total amount of debt transferred to the payment arrangement that has not been canceled.

Clicking **Cancel** causes the system to cancel all related frozen adjustments, returning the debt to the original service agreements. You must specify the cancel reason by selecting it from the drop down list when Payment Arrangement Cancellation window is displayed.

Total Canceled Adj is the total amount of debt transferred to the payment arrangement that has been canceled.

Setting Up Pay Plans

A pay plan (PP) is an agreement with a customer to make specific payments on specific dates (as opposed to payment arrangements where the customer makes fixed payments as part of their regular bills).

Consider the following situations that would require a pay plan:

- A customer makes a request to payoff severely overdue debt in one or more installments. The existence of the pay plan insulates the portion of the debt covered by the plan from C&C processing (specifically, from the account debt monitor).
- A commercial or industrial customer cannot or will not post a deposit. This type of pay plan is not specifically paying off past debt; rather, it can be viewed as an installment plan to prepay a regular service agreement.

The topics in the following section describe how to set up a pay plan.

Contents

- [Pay Plan - Main](#)
- [How To Set Up A Pay Plan](#)
- [Reviewing Pay Plans](#)

Pay Plan - Main

The pay plan transaction allows you to set up a pay plan. When you create a pay plan, the system performs the following functions:

- It creates an active pay plan using the pay plan type entered.
- It defaults the customer as the payor if no 3rd party is defined.

Refer to [The Big Picture Of Pay Plans](#) for information about the financial ramifications of pay plans and for a description of how the system monitors pay plans.

Open **Credit & Collection, Pay Plan** to maintain a customer's pay plans.

Description of Page

Pay Plan and **Pay Plan ID** only appear after the pay plan is added to the database. **Pay Plan ID** is a system-assigned random number that stays with the pay plan for life. **Pay Plan** is a concatenation of important details about the pay plan.

Account ID identifies the account of the customer for which the pay plan exists. This field is protected once the pay plan is **Active**.

The **Status** displays the status of the pay plan. Refer to [The Lifecycle Of A Pay Plan](#) for more information about a pay plan's status.

Created by displays the id of the user who created the pay plan and the date and time on which it was created.

Clicking the **Cancel** button causes the pay plan's status to become **cancelled**.

Last Updated by displays the id of the user who last modified the pay plan and the date and time on which it was changed.

Select a **Pay Plan Type**. The Pay Plan Type defines the **Debt Class** whose debt is insulated by the pay plan. The account's **Current Balance** in this debt class appears beneath as well as the portion that is considered **Delinquent Debt**.

If a 3rd party is responsible for the pay plan's payment, **Third Party Payor** must be checked, and the associated 3rd party selected.

The **Payor Account Id** contains the account ID and name of the person responsible for making the pay plan's payments. If a **Third Party Payor** is responsible, the payor's account appears; otherwise the account whose debt is insulated by the pay plan appears.

The **Start Date** defines the first day on which the pay plan is tracked; it defaults to the current date.

Pay Method defines the manner in which the customer intends to fulfill payments under this pay plan.

Significance of payment method. A pay method can have a grace period associated with it. This period controls when a scheduled payment is considered to be missed. For example, a pay method of "postal" might have a grace period of 3 days. This would give the customer 3 extra days to make the scheduled payments before the pay plan is broken.

In addition to grace period, pay method also control if the system will automatically create an automatic payment on the scheduled payment date. If the pay method so indicates AND the account has been set up for automatic payment, the **PPAPAY** background process will create [Automatic Payments](#) on the scheduled payment date.

The **Scheduled Payments** grid contains the pay plan's scheduled payments. The **Scheduled Date** and **Scheduled Amount** must be entered for each scheduled payment. The **Total Amount** of the scheduled payments is displayed at the bottom.

Free format **Comments** can be entered to describe any special notes about the pay plan.

How To Set Up A Pay Plan

To set up a new pay plan:

- Select the **Account** in question on [Pay Plan - Main](#).

- Select the **Pay Plan Type** for the type.
- Identify a **3rd Party Payor** if there is one.
- Select a **Pay Method**.
- Enter one or more **Scheduled Payments**. The **Total Amount** of the scheduled payments should cover the customer's **Delinquent Debt** if you want the pay plan to protect the customer from additional credit and collection activity.
- Click the **Save** button.

Reviewing Pay Plans

In addition to using this maintenance page for reviewing pay plans, Control Central has a pay plan tree that can be used to view pay plan history, including current status. Refer to [Control Central Pay Plan Tree](#) for more information.

Financial Transactions

In this section, we describe the financial transactions generated as a result of your bills, payments and adjustments.

Contents

- [The Big Picture Of Financial Transactions](#)
- [Financial Transaction](#)
- [Account Financial History](#)
- [Account Bill / Payment History](#)
- [SA Financial History](#)
- [SA Non-Accrual Accounting Balance](#)
- [Balance Control](#)
- [Match Event](#)

The Big Picture Of Financial Transactions

The topics in this section provide background information about a variety of financial transaction issues.

We strongly recommend familiarizing yourself with the topics described in [The Financial Big Picture](#) to fully appreciate the system's financial architecture.

Contents

- [Bill Segment Financial Transactions](#)
- [Payment Segment Financial Transactions](#)
- [Adjustment Financial Transactions](#)
- [Financial Transactions And Aged Debt](#)
- [Current Balance versus Payoff Balance](#)
- [The Source Of GL Accounts On Financial Transactions](#)
- [Obscure Things That Can Happen](#)
- [The Big Picture of Balance Control](#)
- [The GL Interface](#)

Bill Segment Financial Transactions

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

If a bill segment is cancelled, another financial transaction is created to reverse the original financial transaction. The cancellation financial transaction appears on the next bill produced for the account as a bill correction.

For more information about bill segment financial transactions, refer to [Bill Details](#).

Payment Segment Financial Transactions

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

If a payment segment is cancelled, another financial transaction is created to reverse the original financial transaction. The cancellation financial transaction appears on the next bill produced for the account as a negative payment.

For more information about payment segment financial transactions, refer to [Payment Details](#).

Adjustment Financial Transactions

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

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

For more information about adjustment financial transactions, refer to [Adjustment Details](#).

Financial Transactions And Aged Debt

When a financial transaction's related bill segment / payment segment / adjustment is frozen, the financial transaction (FT) is also frozen. When an FT is frozen, its service agreement's debt is impacted. It is important to stress the following in respect of this impact:

- The FT's GL details will be interfaced to the GL when the GL interface next executes.
- If the FT decreases the amount of debt, the customer's aged debt is affected immediately regardless of whether the FT appears on a bill.
- If the FT increases the amount of debt, the amount the customer owes from an aged debt perspective may or may not be affected by the FT. There is a switch on an FT called New Charge that controls the arrears behavior. If this switch is on, the customer's aged debt will not reflect the FT amount until the FT is swept onto a bill. The moment the FT is swept onto the customer's bill, the debt starts aging. If this switch is off, the date on which the FT starts aging must be defined in the Arrears Date field.
- The amount a customer owes in total is immediately affected by the FT regardless of whether the FT appears on a bill. This means that the amount of aged debt may not be in sync with the total amount owed. This seems odd, but is useful from a credit and collections perspective. You see, you probably don't want to start aging a FT until the customer has actually seen it.

For more information, refer to [Financial Transactions Created Between Bills](#).

Current Balance versus Payoff Balance

For information about current and payoff balance in general, refer to [Current Amount versus Payoff Amount](#).

The topics in this section describe when payoff amount differs from current amount for the various types of financial transactions.

Contents

- [Adjustments - Current Balance versus Payoff Balance](#)
- [Billing - Current Balance versus Payoff Balance](#)
- [Payment - Current Balance versus Payoff Balance](#)

Adjustments - Current Balance versus Payoff Balance

For information about current and payoff balance for adjustments, refer to [Adjustments - Current Balance versus Payoff Balance](#).

Billing - Current Balance versus Payoff Balance

For information about current and payoff balance for bill segments, refer to [Billing - Current Balance versus Payoff Balance](#).

Payment - Current Balance versus Payoff Balance

For information about current and payoff balance for payment segments, refer to [Payment - Current Balance versus Payoff Balance](#).

The Source Of GL Accounts On Financial Transactions

For information about the source of the distribution codes used to generate the GL accounts on the financial transactions, refer to [The Source Of GL Accounts On Financial Transactions](#).

Obscure Things That Can Happen

The topics in this section provide information about obscure things that can happen when a financial transaction (FT) is frozen.

Contents

- [A Stopped Service Agreement May Be Closed](#)
- [A Closed Service Agreement May Be Reactivated](#)
- [A Write-Off Process May Be Deactivated If The SA Is Closed](#)
- [A Reactivated Service Agreement May Be Closed](#)

A Collection Process May Be Canceled When A Credit FT Is Frozen
A Severance Process May Be Canceled When A Credit FT Is Frozen
One Or More Algorithms May Be Executed

A Stopped Service Agreement May Be Closed

If an FT causes a **Stopped** SA's current and payoff balances to become zero, the system closes the service agreement (i.e., the service agreement's status becomes **Closed**).

A Closed Service Agreement May Be Reactivated

After a service agreement is closed (i.e., after it's stopped and paid-in-full), it's possible for some types of financial transactions to be linked to the service agreement. These financial transactions could cause the current and/or payoff balances to become non-zero. If this happens, the system reactivates the service agreement (i.e., the service agreement's status becomes **reactivated**).

When a service agreement becomes reactivated, it becomes eligible for review by the write-off process monitor (when it next runs). The write-off process monitor will, in all likelihood, start a write-off process for the reactivated service agreement.

Financial events that can be linked to a closed service agreement are:

- The cancellation of a bill segment
- The freezing of a payment segment
- The cancellation of a payment segment
- The freezing of an adjustment
- The cancellation of an adjustment

A Write-Off Process May Be Deactivated If The SA Is Closed

When a service agreement becomes **Closed**, it will be removed from any ongoing write-off processes. This removal could cause the write-off process to become **Inactive** (if this is the last service agreement linked to the write-off process). The inactivation of the write-off process, could, in turn, cause collection agency referrals associated with the write-off process to be cancelled.

A Reactivated Service Agreement May Be Closed

After a service agreement has been reactivated (refer to the previous section for how this happens), a financial transaction (or transactions) may be linked to it that will cause the current and payoff balance to return to zero. If this happens, the system closes the service agreement (i.e., the service agreement's status becomes **Closed**).

All types of financial transactions can be posted to a **Reactivated** service agreement.

A Collection Process May Be Canceled When A Credit FT Is Frozen

Periodically, the system determines if an account's debt violates your collection criteria. If so, a collection process is created using the violated criteria's collection process template.

The Collection Process Monitor reviews an active collection process whenever one of its service agreements' debt is reduced. Financial events that can cause service agreement debt to be reduced are:

- The cancellation of a bill segment
- The freezing of a payment segment
- The freezing of an adjustment that credits a service agreement

At review time, the Collection Process Monitor removes any service agreement from a collection process when the service agreement has no debt older than the arrears date on the collection process. When all service agreements are removed, the collection process is cancelled.

Refer to [The C&C Monitors](#) for more information about debt monitoring.

A Severance Process May Be Canceled When A Credit FT Is Frozen

If the customer does not respond to a collection process's prodding, the system creates a severance process for each delinquent service agreement associated with the collection process.

The system reviews an active severance process whenever its service agreements' debt is reduced. Financial events that can cause service agreement debt to be reduced are:

- The cancellation of a bill segment
- The freezing of a payment segment
- The freezing of an adjustment that credits a service agreement

At review time, the system determines if the service agreement has no debt older than the arrears date on the severance process. If so, the system looks at the severance process' severance process template:

- If the severance process template indicates that the process can be **Used to sever service**, the system cancels all pending severance events and deactivates the severance process.
Note: the only severance processes that are not used to sever service are those that request reconnection of a cut service.
- If the canceled process has field activities,
 - If the field activities are not linked to a field order and are still in the pending status, the system cancels the field activities.
 - Otherwise, the system generates worklist entries to alert an operator that field activities have been dispatched. The operator must determine how to handle these activities because they could already be scheduled for dispatch (or completed!).

Real time cancellation. Unlike collection processes, the system cancels severance processes real time (i.e., there is no equivalent of the collection process monitor for severance processes). Why are severance processes canceled real time? Because a severance process may have events that create field activities to sever service. These events need to be canceled the moment the FT is frozen, we can't wait until a background process runs. This means that if a customer pays in person for a service agreement that is pending severance, the system will cancel the process and its field activities (if any) the moment the payment is entered.

Refer to [The C&C Monitors](#) for more information about debt monitoring.

One Or More Algorithms May Be Executed

If the FT is linked to a service agreement whose SA type has Freeze Algorithms, these algorithms will be executed when the FT is frozen. Because you can code whatever you please in an algorithm, anything could happen. Examples of such algorithm include:

- An algorithm that routes consumption to third party service providers when a bill is frozen (refer to [STG SND CONS](#) for more information). This type of algorithm would be plugged in on SA types whose consumption can be interfaced to other service providers.
- A bill message can be added to the account when the system detects that its deposit has been paid in full (refer to [DEP PIF MSG](#) for more information). This type of algorithm would be plugged in on deposit SA types.

In addition, FT freeze algorithms can also be defined on an account's customer class. Because you can code whatever you please in an algorithm, anything could happen. An example of such an algorithm is one that cancels match events when an FT is cancelled (refer to [CNCL FT MEVT](#) for more information). This type of algorithm would be plugged in on customer classes associated with open item customers.

The Big Picture of Balance Control

The balance control processes are used to check the financial integrity of your system. The contents of this section describe how these processes work.

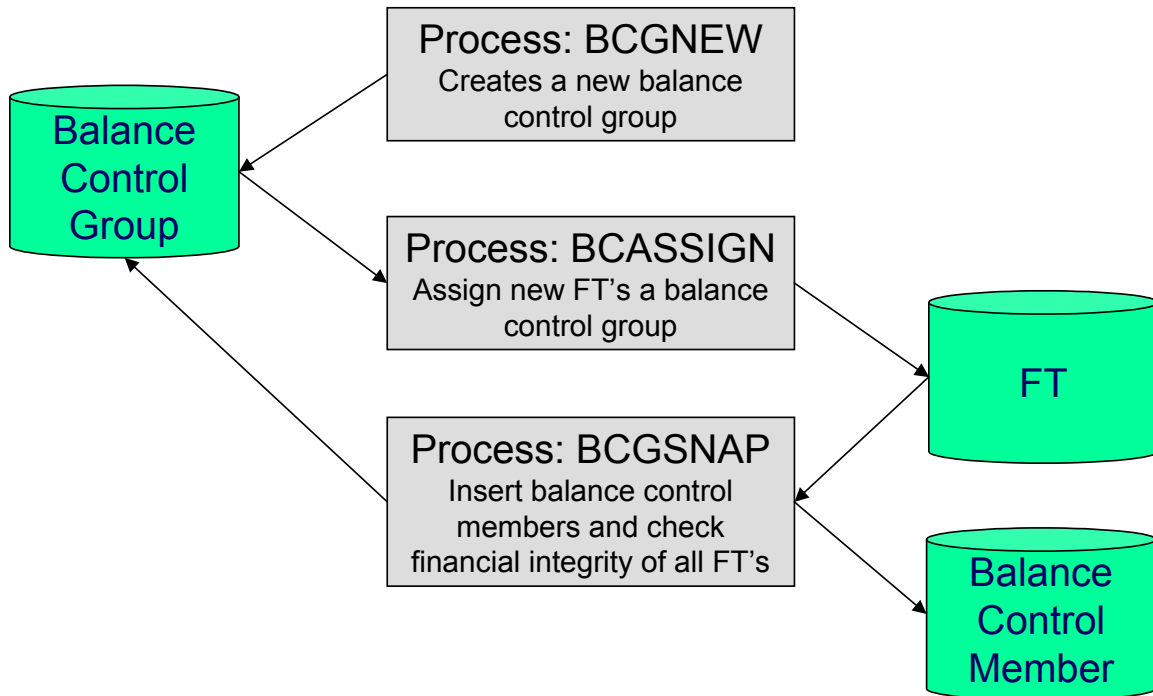
Contents

[The Balance Control Background Processes](#)

[Balance Control Information Is Available Online](#)

The Balance Control Background Processes

The following diagram illustrates the balance control background processes:



Contents

[BCGNEW - Create A New Balance Control Group](#)

[BCASSIGN - Assign New Financial Transactions A Balance Control Group](#)

[BCGSNAP - Insert BC Members And Check Financial Integrity Of All FTs](#)

[Consider A Backup At This Point](#)

BCGNEW - Create A New Balance Control Group

This process creates a **Pending** balance control group if one doesn't already exist. You may wonder why an entire process is dedicated to such a trivial task. The reason is because the next process, BCASSIGN, is a multi-threaded (i.e., parallel) process and we only need one **Pending** balance control group regardless of the number of threads used to assign balance control ID's to financial transactions.

BCASSIGN - Assign New Financial Transactions A Balance Control Group

This multi-threaded (i.e., parallel) process assigns the **Pending** balance control group to new FTs whose freeze date/time is before the create date/time of the balance control record.

BCGSNAP - Insert BC Members And Check Financial Integrity Of All FTs

This process performs the following two functions:

- It summarizes new financial transactions under the current Pending balance control group as follows:
 - It creates a balance control member for every combination of **Division**, **SA Type** and **FT Type** referenced on the financial transactions belonging to the balance control group.
 - It updates each balance control member with the following information:
 - The number of financial transactions (FTs)
 - The sum of the total amounts on the FTs in this balance control group.

- The sum of the current amounts on the FTs in this balance control group.
- The sum of the total amounts from all FTs (in this and all other balance control groups).
- The sum of the current amounts from all FTs (in this and all other balance control groups).
- It sets the status of the balance control group to **Complete**.
- It checks the integrity of the financial transactions in each historical Balance Control Group. It does this by summarizing EVERY financial transaction throughout time and determining if the sums are in sync with the values maintained on the balance control members. If integrity problems are detected, a detailed error message is displayed on the run control associated with the process.

If you opt to run the balance control processes on a nightly basis, you will find that the verification processing will take longer every night (because there are more financial transactions over time). In order to deal with this issue, the [BCGSNAP](#) process has a parameter that allows you to control which of the above functions is implemented (it's call VERIFY-ONLY-SW). In order to speed nightly processing, run this process with the switch set to "G" (this causes new financial transactions to be summarized under a new balance control). Then, once a week (or month), run this process with the switch set to "Y" (this checks the financial integrity of all financial transactions in the system).

If you run the balance control processes less frequently, you can set the VERIFY-ONLY-SW to "N" (this causes both of the above functions to execute).

Consider A Backup At This Point

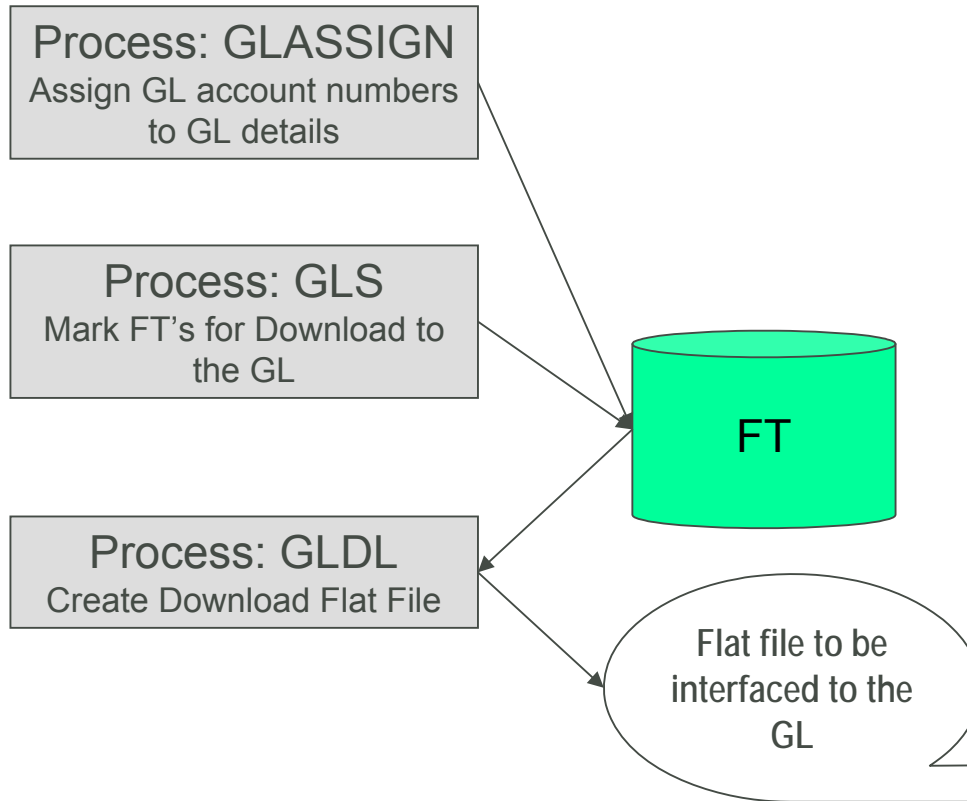
We recommend you backup your database AFTER you run the above processes. Why? So that if a financial integrity problem is spotted in the future, you can compare the current database against the backup to see what changed.

Balance Control Information Is Available Online

Your internal auditors may be interested in the total number and amount of financial transactions that have been posted since a given point in time. You can use the [Balance Control](#) page to see this information.

The GL Interface

The following diagram illustrates the GL Interface.

**Contents**

- [GLASSIGN - Assign GL Account Numbers To GL Details](#)
- [GLS - Prepare FTs for Download](#)
- [GLDL - Create General Ledger Download Flat File](#)

GLASSIGN - Assign GL Account Numbers To GL Details

The **GLASSIGN** process assigns GL account numbers to the GL details associated with financial transactions. GL account numbers are assigned as follows:

- Every GL detail references a distribution code.
- Every distribution code references a GL assignment algorithm. The base package algorithm simply uses the default GL account associated with the distribution code. However, you can construct your own algorithm(s) to assemble your GL account numbers in your desired fashion. Refer to [Setting Up Distribution Codes](#) for more information about the GL format algorithm.
- The **GLASSIGN** process simply calls each GL's details distribution code's GL assignment algorithm and updates the GL detail with the result (i.e., the GL account number). This GL account number is then used by the **GLDL** process when it creates the consolidated journal entry that's interfaced to the GL.

If incorrect GL account numbers get assigned to GL details... If you do not plug in the correct algorithm or your algorithm is wrong, you can correct the GL account numbers that are assigned to the GL details. How? Write a simple program that resets the GL account numbers on the erroneous GL details. Then run **GLASSIGN**. **GLASSIGN** will re-execute the distribution code GL assignment algorithm and refresh the account number accordingly.

GLS - Prepare FTs for Download

The **GLS** process creates FT download staging records for all FTs that are ready to be posted to the GL (the FT download staging records are stored on the FT/Process table). Each FT download staging record is marked with a batch process ID and run number.

- The batch process ID is the process responsible for creating the flat file that contains the consolidated journal entry that is interfaced to your general ledger. The batch process ID is defined on [Installation Options – Financial](#). The system comes with a single GL download program ([GLDL - Create General Ledger Download Flat File](#)). The GL account numbers are provided by the GL account algorithms that are specified on the distribution codes. Refer to [Setting Up Distribution Codes](#) for more information about the GL account algorithm.
- The run number is the batch process ID's current run number.

This process also changes the status of the FT to **distributed**. You may not change the FT's GL details after this time.

GLDL - Create General Ledger Download Flat File

The **GLDL** process creates the flat file that contains the consolidated journal entry that is interfaced to your general ledger. One header and multiple detail records are created as described below. At the conclusion of the batch process, a validation is performed to compare debits against credits. If they are not equal, the process terminates with an error.

Contents

[Header Record Layout](#)

[Detail Record Layout for GLDL](#)

Header Record Layout

Field Name	Format	Source/Value/Description
REC_TYPE	A1	1 (1 is used for header records)
BATCH_CD	A8	The code for the batch process that created the file. (This value will always be 'GLDL'.)
BATCH_NBR	N10	The batch process will be run many times. This field indicates which of those runs produced a particular output file. This may be thought of as an instance identifier for the batch process.
BATCH_RERUN_NBR	N10	Any given instance of the GLDL batch process may be re-run at any time. If this instance has been re-run, this field will be populated with a number indicating how often.
EXTRACT_DTTM	A26	The date and time of the run that created a particular output file.
DETAIL_REC_CNT	N12	The number of details records
DETAIL_REC_TOTAL_DR	N13.2	This is the sum of FINANCIAL_AMOUNTs from all detail records

Field Name	Format	Source/Value/Description
		that were greater than zero (debits).
DETAIL_REC_TOTAL_CR	N13.2	This is the sum of FINANCIAL_AMOUNTs from all detail records that were less than zero (credits).

Detail Record Layout for GLDL

Note. A record is created for each unique occurrence of REC_TYPE, GL_DIVISION, CURRENCY_CD, GL_ACCOUNT and ACCT_PERIOD. If a given GL account has debit and credit FINANCIAL_AMOUNTs, two records will be created – one shows the debit amount, the other shows the credit amount.

Field Name	Format	Source/Value/Description
REC_TYPE	A1	2 (2 is used for detail records)
GL_DIVISION	A5	This is the GL Division from the CI_FT record. It determines a) which Accounting Calendar will be used to calculate the Accounting Period (below) and b) the currency of the FT.
CURRENCY_CD	A3	The currency in which the Amount is denominated.
GL_ACCOUNT	A254	Contains the GL account number as supplied by the distribution code's GL account algorithm .
ACCT_PERIOD	A6	An accounting period in the format YYYYPP where YYYY is the fiscal year and PP is the accounting period. This is derived from the GL detail's FT's accounting date and GL division.
FINANCIAL_AMOUNT	N13.2	This is the debit or credit amount. Debits are represented by positive numbers; credits are represented by negative numbers. If a given GL account has debit and credit entries, two records will be created – one shows the debit amount, the other shows the credit amount.
STAT_CODE	A8	This is the GL distribution code's statistic code (if any)
STAT_AMOUNT	N13.2	This is the statistical amount from the GL detail lines

Note. The GLASSIGN process updates an FT's GL details with the appropriate GL account number.

Financial Transaction

Payment segments, adjustments and bill segment have a corresponding financial transaction. The financial transaction is created by the system when any of the source transactions is created. It contains the financial effects of its corresponding adjustment / bill segment / payment segment.

For background information, refer to [The Big Picture Of Financial Transactions](#).

The topics in the section describe the financial transaction page.

Contents

[Financial Transaction - Main](#)

[Financial Transaction - FT Process](#)

Financial Transaction - Main

Use **Financial, Financial Transaction** to update and maintain financial transaction information.

Navigating from source transactions. On the bill, payment, and adjustment maintenance pages, you can click the financial transaction go to button to transfer to this page.

Description of Page

Most of the attributes on this page are display-only. The following points describe the conditions under which certain fields may be modified:

- **Accounting Date** may be modified until the financial transaction (FT) is interfaced to the GL (i.e., until the **GL Distribution Status** is *Distributed*).
- **New Charge** may be modified until the FT is swept onto a bill (when the next bill is completed for the FT's account). If **New Charge** is modifiable and turned off, **Arrears Date** must be specified.
- **Show on Bill** and **Correction** can be modified until the FT is frozen.

The remainder of this section defines each of the fields on the page.

FT Type is the type of financial transaction. Values are: *Adjustment*, *Adjustment Cancellation*, *Bill Segment*, *Bill Segment Cancellation*, *Pay Segment*, and *Pay Segment Cancellation*.

Click the go to button to view the originating transaction on the adjustment, bill, or payment page.

FT ID is the system-assigned, unique identifier of the financial transaction.

SA ID is the system-assigned, unique identifier of the service agreement to which the financial transaction is linked.

Bill ID is the system-assigned, unique identifier of the bill to which the financial transaction is linked. This field is only populated after the FT is swept onto a bill (and this happens when the next bill is completed for the FT's account).

Sibling ID is the system-assigned, unique identifier of the source transaction associated with the FT.

- If this FT is associated with a bill segment, **Sibling ID** contains the unique identifier of the bill segment.
- If this FT is associated with a payment segment, **Sibling ID** contains the unique identifier of the payment segment.
- If this FT is associated with an adjustment, **Sibling ID** contains the unique identifier of the adjustment.

Parent ID contains the following:

- If this FT is associated with a bill segment, **Parent ID** contains the unique identifier of the bill with which the bill segment is associated.

- If this FT is associated with a payment segment, **Parent ID** contains the unique identifier of the payment with which the payment segment is associated.
- If this FT is associated with an adjustment, **Parent ID** contains the adjustment type.

Create Date/Time are the date and time the FT was created.

Accounting Date is the accounting date that will be used by the general ledger to define the accounting period(s) into which the FT will be booked.

CIS Division is the CIS division associated with the FT. This comes from the CIS division linked to the FT's service agreement's SA type.

GL Division is the GL division associated with the FT. This comes from the GL division linked to the FT's service agreement's SA type.

Show on Bill indicates if the FT will be shown on the customer's bill. You should only turn this off for erroneous FTs that should not be shown to the customer. For example, if you cancel / rebill a bill segment and you want to suppress the resultant FTs on the printed bill, turn this switch off.

New Charge indicates if the FT's charge only starts aging when the FT is swept onto the next bill produced for the account. If you turn this switch off, you must enter the date the FT starts aging in **Arrears Date**.

Not In Arrears indicates if the FT's financial impact has been canceled and therefore should not be considered for arrears purposes. This switch is turned on by the system when a FT's source transaction in canceled (both the original FT and the cancellation FT are marked as **Not In Arrears**).

Match Event ID is the system-assigned, unique identifier of the match event to which the financial transaction is linked. This field is only enabled for [Open Item Accounts](#). Be aware that changing a financial transaction's match event can result in the balancing / unbalancing of the prior and newly referenced match events.

Arrears Date is the date the FT starts aging. This field is typically blank until the FT is swept onto the account's bill. If you want to start aging an FT on an historical date (for whatever reason), enter the appropriate **Arrears Date**. (Note - you must uncheck the "New Charge" box first in order to do this.)

Correction causes the FT to be summarized in the correction area of the bill-at-a-glance. This is set by the system automatically when a bill segment is cancelled / rebilled after its parent bill is completed (i.e., sent to the customer).

Redundant indicates if the FT's financial impact is considered irrelevant. This only happens after an FT has reached an age that is no longer relevant for aging purposes (e.g., when the FT is older than 120 days – or whatever age has been set as the Oldest Bucket Age on the Installation Options) and when its balance is exactly equal to zero with respect to other redundant FTs of the service agreement.

Transferred Out indicates if the FT has been transferred to a service provider. This could only happen if the FT is associated with a SA that is associated with a "they bill for us" service provider. Refer to [They Bill For Us](#) for more information.

The **Frozen** switch is turned on when the FT has been frozen (i.e., posted to the service agreement's payoff and/or current balances). If this switch is on, both **Freeze Date/Time** and **Frozen By** are populated.

Current Amount contains the FT's impact on the service agreement's current amount.

Payoff Amount contains the FT's impact on the service agreement's payoff amount.

For more information, refer to [Current Balance versus Payoff Balance](#).

Currency Code is the currency code associated with the FT's account.

GL Distribution Status defines the status of the FT in respect of its interface to the general ledger. When an FT is first created, its status is **Pending**. When an FT is frozen, this value is set to **Generated**. If you modify the GL details, the status becomes **Modified**. After it has been marked for interface to the general ledger by the **GLS** process, the status becomes **Distributed**.

GL Extract Dates display the **Scheduled** date that the GL details will be marked for interface to the general ledger. Note that this date is typically only set to a future date for FTs associated with automatic payments. **Actual** displays the date that the GL details were actually posted to the general ledger.

The grid at the bottom shows the FT's debits and credits (i.e., the detail journal lines). Debits are shown as positive amounts, credits are negative amounts. The following points describe rules governing if and how the information in the grid can be modified:

- The FT GL details may not be modified if the FT is **Pending** or **Distributed**.
- All **Amounts** must sum to zero.
- Multiple FT GL details may be set to affect the **Total Amount** (although they are normally generated with one GL checked to affect total amount).
- All FT GL's checked to affect **Total Amount** must sum to the **Payoff Amount** of the FT.

The following information is displayed:

- **Sequence number** is the system-assigned, unique identifier of the journal line.
- **Total Amount** defines if the journal line contains the total of the other journal lines. For example, on a bill segment for utility service, this would be turned on for the receivable account because its amount equals the sum of the other payable and revenue GL accounts.
- **Distribution Code** is the CIS distribution code from which the GL account constituents are derived.
- If the **GL Account** has been populated, the GL account number is displayed.

Refer to [GLASSIGN - Assign GL Account Numbers To GL Details](#) for more information about how the GL account is populated.

- **Amount** defines the journal line's amount.
- **Statistic Amount** defines the statistical amount that will be posted to the GL. This value is only populated on distribution lines created for rate components designated as affecting GL statistical quantity.
- **Characteristic Type** and **Characteristic Value** describe the characteristic value that was used when the line's amount was calculated. This information is only displayed if the journal line was derived from bill calculation lines that were calculated using a bill factor (because only bill factor's use characteristic values). Refer to [An Illustration Of A Bill Factor And Its Characteristics](#) for more information.

Tax reporting. It is important to note that a journal line's characteristic value is NOT interfaced to the GL. We have included this attribute on the journal line so that tax reporting can be performed from this system (tax reporting typically necessitates showing each taxing authority - the characteristic value - that participated in a given tax payable GL account).

- If you have configured your installation options to indicate that [fund accounting](#) is **practiced**, the description of the **Fund** associated with this distribution code is displayed.

Financial Transaction - FT Process

Use **Financial, Financial Transaction, FT Process** to view those ancillary processes that may be triggered as a result of this financial transaction.

Algorithms cause financial transactions to be linked to batch processes. Financial transactions get associated with a given process / batch number when certain algorithms are executed. For example, when a bill segment is frozen, the system executes the freeze algorithms associated with the bill segment's service agreement's SA type. One of these algorithms may check if there are service providers who are interested in the bill segment's consumption, and if so, schedule the consumption to be downloaded by linking the bill segment's financial transaction to the service provider's consumption download process.

Description of Page

This page shows the **Batch Processes** associated with a given **Financial Transaction**. Information on this page may not be modified. This information appears purely for audit purposes.

Account Financial History

This page shows how an account's current and payoff balance have changed over time. Use **Financial Query, Account Financial History** to open this page.

Description of Page

This page is dedicated to a grid that shows an account's financial events. These events are grouped together by Arrears Date, Financial Transaction Type and Parent Id (therefore if there were two payments on the same date, two rows would appear).

Multiple adjustments of the same type on the same date. If multiple adjustments with the same adjustment type exist on the same date, their total amount will appear as a single row on this query.

You can use this grid to both view high-level information about these events and to transfer to the respective page in which an object is maintained. The following columns are displayed in the grid:

Arrears Date

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

Financial Transaction Type	This column indicates the type of financial event: <i>Bill Segment</i> , <i>Pay Segment</i> , <i>Bill Segment Cancellation</i> , <i>Pay Segment Cancellation</i> , <i>Adjustment</i> and <i>Adjustment (Cancel)</i> . If the event is related to an adjustment, the adjustment type's description is displayed instead of "Adjustment".
Current Amount	This column shows the financial event's effect on the account's current balance.
Current Balance	This column shows the account's current balance after the financial event.
Payoff Amount	This column shows the financial event's effect on the account's payoff balance. The value is grayed out if it is the same as the current amount.
Payoff Balance	This column shows the account's payoff balance after the financial event. The value is grayed out if it is the same as the current balance.

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

For information about current and payoff balance, refer to [Current Amount versus Payoff Amount](#).

Account Bill / Payment History

This page shows an account's bills and payments. Use **Financial Query, Bill / Payment History** to open this page.

Warning! For [balance-forward accounts](#), bill rows contain the balance presented on the respective bill, and payment rows contain the amount of the respective payment. However, for [open-item accounts](#), this query behaves differently - see the description of page below for the details.

Description of Page

This page is dedicated to a grid that shows the account's bills, credit notes, correction notes, payments and payment cancellations. You can use this grid to both view high-level information about these objects and to transfer to the respective page on which an object is maintained.

The area beneath **Account ID** provides you with options that control which transactions appear in the grid. The following points describe the various options:

- Use **Transaction Type Filter** to restrict the type of transactions that appear in the grid. The following options are available:
 - **All**. This option shows all transactions.
 - **Bill**. This option shows all bills.

- **Credit Note.** This option shows all credit notes.
- **Correction Note.** This option shows all correction notes.
- **Not Billed Yet.** This option shows a single line with a summary of frozen financial transactions that have not appeared on a bill yet.
- **Payment.** This option shows all payments.
- **Payment Cancellation.** This option shows all payment cancellations.

Credit Notes or Correction Notes. The Bill Correction option on the Installation table controls whether Credit Notes or Correction Notes are allowed. The default transaction type filter displayed is for Credit Note. If your implementation uses Correction Notes, you'll need to change the transaction type filter to reflect this. This is done by overriding the label for lookup value **CRNT** on the customizable lookup field **TXN_FLTR_TYPE_FLG**.

- Use **Match Event Status Filter** to restrict the transactions based on the status of their [match event](#). This filter only appears if the bill's account is an [open-item](#) customer. The following options are available:
 - **All.** This option shows all transactions regardless of the status of their match events.
 - **Balanced.** This option shows all transactions with at least one **balanced** match event.
 - **Disputed.** This option shows all transactions with at least one **disputed** match event.
 - **Unbalanced.** This option shows all transactions with at least one **unbalanced** match event.
 - **Unmatched.** This option shows all transactions with at least one financial transaction that is not linked to a match event.
- Use **Date Range From** and **To** to restrict the transactions based on arrears date.

Don't forget to click the search button after changing the filters or after selecting a new Account ID.

For [balance-forward accounts](#), bill rows contain bill information including the balance presented on the respective bill, and payment rows contain the amount of the respective payment.

For [open-item accounts](#), the grid behaves differently:

- The amount on bill rows is equal to the sum of the current charges, adjustments and corrections on the bill. Payment rows contain the amount of the respective payment.
- Credit notes appear as separate rows and contain the amount of canceled bill segments represented on the credit note.
- Correction notes appear as separate rows and contain the difference between the canceled bill segments amount and the rebilled bill segments amount represented on the correction note.
- Each row contains an indication if all of its financial transactions are fully matched.
- A summary of the match status of its financial transactions is shown in the adjacent columns:
 - **Balanced** contains the count and total amount of financial transactions linked to this activity that are linked to **balanced** match events.

- **Unbalanced** contains the count and total amount of financial transactions linked to this activity that are linked to **unbalanced** match events.
- **Disputed** contains the count and total amount of financial transactions linked to this activity that are linked to **disputed** match events.
- **Unmatched** contains the count and total amount of financial transactions linked to this activity that are not linked to any match event.

You can use the hyperlinks to view the detailed financial transactions that are summarized in each cell.

SA Financial History

This page is dedicated to a grid that shows the financial transactions linked to a service agreement. Use **Financial Query, SA Financial History** to open this page.

Description of Page

This page is dedicated to a grid that shows a service agreement's financial transactions (FT). You can use this grid to both view high level information about these objects and to transfer to the respective page in which an object is maintained.

The following columns are displayed in the grid:

Arrears Date	This is the date the FT starts aging. This column will be blank if the FT has not started aging yet.
Financial Transaction Type	This column indicates the type of financial event: Bill Segment , Pay Segment , Bill Segment Cancellation , Pay Segment Cancellation , Adjustment and Adjustment (Cancel) . If the event is related to an adjustment, the adjustment type's description is displayed instead of "Adjustment".
Current Amount	This column shows the FT's effect on the service agreement's current balance.
Current Balance	This column shows the service agreement's current balance after the financial event.
Payoff Amount	This column shows the FT's effect on the service agreement's payoff balance. The value is grayed out if it is the same as the current amount.
Payoff Balance	This column shows the service agreement's payoff balance after the financial event. The value is grayed out if it is the same as the current balance.

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

For information about current and payoff balance, refer to [Current Amount versus Payoff Amount](#).

SA Non-Accrual Accounting Balance

Only relevant if you practice non-accrual accounting. This transaction is only relevant if you practice non-accrual accounting (e.g., you only pay the taxing authorities when the customer pays you or you defer revenue recognition till a later date or a combination of the two). Refer to [Payables Cash Accounting](#) and [Deferred Accrual Accounting](#) for more information about non-accrual accounting.

This page displays a grid that shows a service agreement's non-accrual accounting balance for each non-accrual holding distribution code that's been booked to. Use **Financial Query, SA Non-Accrual Accounting Balance** to open this page.

Description of Page

The following columns are displayed in the grid:

Holding Distribution Code	This is a general ledger distribution code used as the holding account for non-accrual accounting. Each holding account that's been used by this service agreement will appear here.
Payable Distribution Code	This is the general ledger distribution code to which the payable is (or will be) transferred when the cash event occurs or when the designated revenue recognition date elapses. The system supports revenue recognition on the bill due date.

For more information on how the distribution codes are set up, refer to [Setting Up Distribution Codes](#).

Payoff Balance	This column shows the service agreement's balance for each non-accrual holding account. If this amount is non-zero, it indicates that for this service agreement we are still holding some payables back until payment is received by the customer, at which time some or all of this amount will be transferred to the payable distribution code.
Accounting Method	This column shows which accounting method is used for the distribution.
Accounting Priority	This column shows the priority level associated with the distribution code. If more than one distribution code is to receive a payable then the priority level, along with debt age, will determine the order in which the transfers are made.

For information about **non-accrual** accounting, refer to [Payables Cash Accounting](#) and [Deferred Accrual Accounting](#).

Balance Control

This page is used to summarize the financial transactions that belong to a particular balance control group.

Refer to [The Big Picture of Balance Control](#) for more information.

Balance Control information is current only as of the Create Date/Time listed. The information displayed on the page is captured when the balance control background process is run. Any financial activity subsequent to the create date and time will not be shown.

Use **Financial Query, Balance Control** to open this page.

Description of Page

This page displays summarized financial information for a particular balance control group. It also details financial information for all SA Types that belong to the group.

The following fields are displayed on the page:

Group ID	This is a sequential value that uniquely identifies a particular balance control run. The Group ID is assigned and incremented automatically every time the Balance Control background process is run.
Status	This is the status of the balance control record: Pending , and Complete . The status of the balance control record will only be pending while the balance control background process is being executed. Balance control information is only reliable if the status is complete.
Create Date/Time	This is the date and time that the balance control record was created. Financial information displayed on this page will only contain information for financial transactions frozen before this date and time.
Current Amount	This is the sum of the current amounts of all FTs that belong to the balance control group.
Payoff Amount	This is the sum of the payoff amounts of all FTs that belong to the balance control group.

Current Balance This is the sum of the current amounts of all FTs that belong to this balance control group plus all earlier balance control groups. Therefore this is the instantaneous current balance of the entire CIS as of the create date and time.

Payoff Balance This is the sum of the payoff amounts of all FTs that belong to this balance control group plus all earlier balance control groups. Therefore this is the instantaneous payoff balance of the entire CIS as of the create date and time.

The number of FT information details the number of bill segments, pay segments, adjustments and their cancellations that belong to the balance control group.

Adjustment The total number of adjustments that belong to this balance control group.

Adjustment Cancellation The total number of adjustment cancellations that belong to this balance control group.

Bill Segment The total number of bill segments that belong to this balance control group.

Bill Segment Cancellation The total number of bill cancellations that belong to this balance control group.

Pay Segment The total number of pay segments that belong to this balance control group.

Pay Segment Cancellation The total number of pay cancellations that belong to this balance control group.

The **SA Type** scroll summarizes the FTs that belong to a particular SA Type within the balance control group – otherwise all fields are defined as above. You can scroll through all SA Types that belong to the balance control group.

SA Type The SA Type being summarized.

Match Event

Warning! Match Events are only used if you practice [Open Item Accounting](#).

Match events are used to match debit and credit financial transactions together. When financial transactions are linked to a **balanced** match event, they no longer affect the customer's arrears.

You can use the match event page to do the following:

- Add, change, delete, and cancel match events.
- Add and remove financial transactions from / to a match event.
- Designate financial transactions as being "in dispute" (disputed financial transactions do not affect aged debt until they are resolved).

- View all unmatched financial transactions linked to an account.

Refer to [Match Events](#) for a full description of the lifecycle of a match event and a description of how the system automatically creates match events.

Contents

- [Match Event - Main](#)
- [Match Event - FT Details](#)
- [Match Event - Subtotals](#)
- [How To Perform Common Match Event Functions](#)

Match Event - Main

A typical match event matches a bill's financial transactions with a payment's financial transactions. The **Main** page allows you to define the bill(s) and payment(s) whose financial transactions are matched together. Use **Financial, Match Event** to open this page.

Debit must equal credits. Before a match event impacts a customer's arrearage, its debits and credits must net to zero for every service agreement referenced on the match event. Until that time, the financial transactions on a match event continue to affect arrearage. The only exception is the case of [disputes](#).

You can match any debit and credit. While the above describes the matching of bills and payments, it's important to remember that a match event matches any type of debit with any type of credit. This means that a match event could match a bill with a credit or correction note, or a payment with a payment cancellation.

You can match any number of payments under a match event. While most match events deal with a single bill and a single payment, there's no limitation to the number of payments on a match event. The only restriction is that the debits and credits must net to zero for all service agreements.

It is better not to mix multiple bills on a single match event. For purposes of bill balance information, it is strongly recommended that you compose your match events with financial transactions limited to a single bill. If you mix financial transactions from multiple bills on a single match event you will not be able to determine the unpaid balance of a partially paid bill.

You can match specific financial transactions. While most match events deal with every financial transaction on a bill and payment, a match event can deal with individual financial transactions. For example, a match event could match a bill segment with a combination of a payment segment and a write-off adjustment. If you need to add or remove a specific financial transaction (i.e., a bill segment, payment segment, or adjustment), navigate to the **FT Details** tab. Perhaps a better way to differentiate between this page and the **FT Details** tab is to consider the example of a bill with 100 bill segments. When you link this bill to a match event, you are actually linking its 100 financial transactions. If you wanted to add only a subset of this bill's financial transactions to the match event, you'd use the **FT Details** tab.

Refer to [How To](#) for instructions describing how to perform common match event maintenance functions.

Description of Page

This page is used to maintain the financial transactions (FTs) that are linked to a match event. The remainder of this section defines each of the fields on the page.

Match Event Info and **Match Event ID** only appear after the match event exists on the database. The **ID** is a system assigned random number that stays with the match event for life. **Match Event Info** is a concatenation of important details about the match event and its account.

The next section contains the sum of the **Debit** and **Credit** financial transactions linked to the match event. If the debits and credits do not sum to zero, the **Difference** is also shown.

Account ID defines the account whose financial transactions are matched under this match event. This field is gray after the match event is added to the database.

Match Event Status defines the state of the match event.

- Match events are initially created in the **open** state. Please note that you may delete an **open** match event.
- The system automatically changes an **open** event's status to **balanced** when the sum of the **Debit(s)** equals the sum of the **Credit(s)** for each service agreement on the match event. It's worth stressing that a match event may contain financial transactions from many service agreements and each service agreement's financial transactions must sum to zero before the match event can become **balanced**.
- You may re**open** a **balanced** event (by adding / removing items so that the match event becomes unbalanced).
- You can cancel a **balanced** or **open** match event by changing the **Match Event Status** to **cancelled**. You must also define a **Cancel Reason** if you cancel a match event. Refer to [How Are Match Events Cancelled?](#) for more information about cancellation.

Turn on **Dispute** if this match event exists to designate certain financial transactions as **disputed**. In addition,

- Describe the reason for the dispute in **Remarks**.
- Link the disputed financial transactions to the match event by selecting the bill(s) below. If only a subset of a bill is disputed:
 - Click on the respective hyperlink in the **Unmatched Debits** column grid (this will transfer you to the next tab with these financial transactions displayed).

- Select the bill segments that you want to designate as disputed.
- Press the **Link / Unlink** button to link the selected bill segments to the match event.

Disputing items on a balanced event. The **Dispute** switch will be protected when the match event is **Balanced** or **Cancelled**. If the items on a **balanced** event are being disputed, you must add / remove items so that the match event becomes **Open** before you can turn on the **Dispute** switch.

The remainder of the page is dynamic depending on the **Match Event Status**:

- If the status is **Balanced** or **Open**, a grid appears containing a summary of the bills, payments, credit or correction notes, and payment cancellations that contribute financial transactions to the match event (note, we refer to these collectively as "contributing objects" in the following discussion). The following columns appear in this grid:
 - **Transaction Type** defines whether the contributing object is a **Bill**, **Payment**, **Payment Cancellation** or **Credit** or **Correction Note**. In the rare situation when unbilled financial transactions are linked to the match event, a transaction type of **Not Billed Yet** appears. If the contributing object is a bill its bill id is also displayed. If sequential bill functionality is enabled, the bill's sequential id is displayed instead.
 - **Matched Activity Information** contains summary information about the contributing object. This column is blank if the transaction type is **Not Billed Yet**.
 - The remaining columns contain the count and amount of each contributing object's financial transactions categorized as follows (note, you can drill down on the count or amount to see the specific financial transactions (FT's) on the next tab).
 - **Matched Debits** summarizes the contributing object's debit FT's that are linked to this match event. A checkbox appears if the count is greater than zero. If you select the checkbox and press the **Link / Unlink** button, these debits will be removed from the match event.
 - **Matched Credits** summarizes the contributing object's credit FT's that are linked to this match event. A checkbox appears if the count is greater than zero. If you select the checkbox and press the **Link / Unlink** button, these credits will be removed from the match event.
 - **Other Debits** summarizes the contributing object's debit FT's that are NOT linked to this match event.
 - **Other Credits** summarizes the contributing object's credit FT's that are NOT linked to this match event.

Adding more financial transactions to a balanced match event. When a match event is **Balanced**, the grid showing unmatched FT's is suppressed (and therefore you can't add additional FT's to the match event). To expose this grid, simply change the status of the match event to **Open**.

- If the status is **Open**, a second grid appears containing a summary of the bills, payments, credit notes, correction notes and payment cancellations with at least one unmatched financial transaction (we refer to these collectively as "unmatched objects" in the following discussion). The following columns appear in this grid:

- **Transaction Type** defines whether the unmatched object is a **Bill**, **Payment**, **Payment Cancellation**, **Credit** or **Correction Note**. A transaction type of **Not Billed Yet** is used for unbilled financial transactions (e.g., adjustments generated between bills). If the unmatched object is a bill its bill id is also displayed. If sequential bill functionality is enabled, the bill's sequential id is displayed instead.
- **Unmatched Activity Information** contains summary information about the unmatched object. This column is blank if the transaction type is **Not Billed Yet**.
- The remaining columns contain the count and amount of each unmatched object's financial transactions categorized as follows (note, you can drill down on the count or amount to see the specific financial transactions (FT's) on the next tab).
 - **Unmatched Debits** summarizes the unmatched object's debit FT's that are not linked to any match event. A checkbox appears if the count is greater than zero. If you select the checkbox and press the **Link / Unlink** button, these debits will be added to the match event.
 - **Unmatched Credits** summarizes the unmatched object's credit FT's that are not linked to any match event. A checkbox appears if the count is greater than zero. If you select the checkbox and press the **Link / Unlink** button, these credits will be added to the match event.
 - **Matched Debits** summarizes the unmatched object's debit FT's that are linked to any match event.
 - **Matched Credits** summarizes the unmatched object's credit FT's that are linked to any match event.

Credit Notes or Correction Notes. The Bill Correction option on the Installation table controls whether Credit Notes or Correction Notes are allowed. The default Transaction Type displayed is for Credit Note. If your implementation uses Correction Notes, you'll need to change the transaction type filter to reflect this. This is done by overriding the label for lookup value **CRNT** on the customizable lookup field **TXN_FLTR_TYPE_FLG**.

The last section contains the running total of the **Debit** and **Credit** financial transactions that have been selected / unselected in the above grids. If debits and credits do not sum to zero, the **Difference** is also shown. These values differ from the values on the top of the page as they are updated when a user selects / unselects an object (whereas the values at the top of the page are only updated when the database is changed).

The **Link / Unlink** button becomes enabled when you select an object in the grids. When you press this button, the financial transactions related to the object are added to / removed from the match event.

Match Event - FT Details

On the **Main** tab, you can add and remove bills, payments, credit notes, correction notes and payment cancellations to / from a match event. When you do this, you are actually adding and removing all of the financial transactions linked to these objects. For example, when you link a bill with 100 bill segments to a match event, you are actually linking its 100 financial transactions.

You need only use the **FT Details** tab when you need to add or remove specific financial transactions. For example, you would use the **FT Details** tab if you need to remove 1 of a bill's 100 financial transactions from a match event.

Use **Financial**, **Match Event**, **FT Details** to open this page (note, you can also open this page using many of the hyperlinks on the Main and SA Subtotals tabs).

Description of Page

This page is used to maintain the financial transactions (FTs) that are linked to a match event. The remainder of this section defines each of the fields on the page.

Match Event Info and **Match Event ID** only appear after the match event exists on the database. The **ID** is a system assigned random number that stays with a match event for life. The **Match Event Info** is a concatenation of important details about the match event and its account.

The next section contains the sum of the **Debit** and **Credit** financial transactions linked to the match event. If the debits and credits do not sum to zero, the **Difference** is also shown.

The following **Filters** work together to restrict the financial transactions that appear in the grid. The following points describe the various options (note, don't forget to press the search button after specifying the various filter options):

- Use **Transaction Type Filter** to restrict the type of transactions that appear in the grid. The following options are available:
 - **All**. Use this option if you do not wish to restrict financial transactions based on their transaction type.
 - **Bill**. This option allows you to view a specific bill's financial transactions. When this option is selected, an input field appears in which you identify the **Bill ID**.
 - **Credit Note**. This option allows you to view a specific credit note's financial transactions. When this option is selected, an input field appears in which you identify the **Bill ID** (every credit note has a unique bill ID).
 - **Correction Note**. This option allows you to view a specific correction note's financial transactions. When this option is selected, an input field appears in which you identify the **Bill ID** (every correction note has a unique bill ID).
 - **Not Billed Yet**. This option allows you to view financial transactions that haven't appeared on a bill yet (e.g., adjustments and corrections).
 - **Payment**. This option allows you to view a specific payment's financial transactions. When this option is selected, an input field appears in which you identify the **Payment ID**.
 - **Payment Cancellations**. This option allows you to view a specific canceled payment's financial transactions. When this option is selected, an input field appears in which you identify the **Payment ID**.

Credit Notes or Correction Notes. The Bill Correction option on the Installation table controls whether Credit Notes or Correction Notes are allowed. The default Transaction Type displayed is for Credit Note. If your implementation uses Correction Notes, you'll need to change the transaction type filter to reflect this. This is done by overriding the label for lookup value **CRNT** on the customizable lookup field **TXN_FLTR_TYPE_FLG**.

- Use **Linkage Filter** to restrict the transactions based on the match event to which they are linked. The following options are available:
 - **All**. This option shows all financial transactions regardless of their match event.

- **Linked to another Match Event.** This option shows financial transactions linked to a different match event.
- **Linked to any Match Event.** This option shows financial transactions linked to any match event.
- **Linked to this Match Event.** This option shows financial transactions linked to this match event.
- **Unmatched.** This option shows financial transactions that are not linked to a match event.

This filter is protected and set to **Linked to this Match Event** if the **Transaction Type Filter** is **All**.

- Use **Debit / Credit Filter** to restrict the transactions based on whether they are debits or credits. The following options are available:
 - **All.** This option shows all debit and credit financial transactions.
 - **Credit.** This option shows all credit financial transactions.
 - **Debit.** This option shows all debit financial transactions.
- Use the **SA Filter** to define the types of service agreements whose financial transactions appear in the grid. The following options are available:
 - **Address.** Use this option to restrict financial transactions to those whose service agreements are 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 financial transactions based on service agreement attributes.
 - **Geographic Type.** Use this option to restrict financial transactions to those whose service agreements are linked to service points associated with a given **Geo Type** and **Value**.
 - **SA ID.** Use this option to restrict financial transactions to those linked to a specific **Service Agreement**.
 - **SA Type.** Use this option to restrict financial transactions to those whose service agreements are linked to a given **CIS Division** and **SA Type**.

Don't forget to click the search button after changing the filters.

The **Select All / Clear All** buttons are used to select financial transactions to add to / remove from the match event (note, after selecting the desired financial transactions, you must also press the **Link / Unlink** button at the bottom of the page).

50 financial transactions at a time. Clicking **Select All** selects the first 50 bill segments in the grid. If more than 50 financial transactions exist, you must select them in batches.

The grid that follows contains the financial transactions (FT) that match your search criteria. The following information appears in the grid:

- **Select box.** Select the FT if you want to Link / Unlink it. You implicitly link FT's that are NOT already linked and you implicitly unlink FT's that are already linked. This field is protected if the FT is linked to another match event.

- **FT Amount.** This column contains the amount of the financial transaction.
- **FT Type.** This column indicates the type of financial transaction: *Bill Segment*, *Bill Segment Cancellation*, *Pay Segment*, *Pay Segment Cancellation*, *Adjustment* and *Adjustment Cancellation*.
- **Arrears Date.** This column contains the financial transaction's arrears date.
- **SA Information.** This column contains a summary of the financial transaction's service agreement.
- **Remarks.** This column highlights the financial transaction's match status: *Linked to this match event*, *Not linked to a match event*, *Linked to another match event* - match event status.
- **Premise Information.** This column contains a summary of the premise (if any) associated with the financial transaction's service agreement.

The last section contains the running total of the **Debit** and **Credit** financial transactions that have been selected / unselected in the grid. If debits and credits do not sum to zero, the **Difference** is calculated. These values differ from the values on the top of the page as they are updated when a user selects / unselects an object (whereas the values at the top of the page are only updated with the database is changed).

The **Link / Unlink** button becomes enabled when you select a row in the grid. When you press this button, the financial transactions related to the object are added to / removed from the match event.

Match Event - Subtotals

Before a match event impacts a customer's arrearage, its debits and credits must net to zero for every service agreement referenced on the match event. This page shows the sum of the debits and credits for every service agreement that contributes at least one financial transaction to the match event. Use **Financial**, **Match Event**, **Subtotals** to open this page

Description of Page

This page shows the sum of the debits and credits for every service agreement that contributes at least one financial transaction to the match event.

Match Event Info and **Match Event ID** only appear after the match event exists on the database. The **ID** is a system assigned random number that stays with a match event for life. The **Match Event Info** is a concatenation of important details about the match event and its account.

The next section contains the sum of the **Debit** and **Credit** financial transactions linked to the match event. If the debits and credits do not sum to zero, the **Difference** is also shown.

The following **Filters** work together to restrict the service agreements that appear in the grid. The following points describe the various options (note, don't forget to press the search button after specifying the various filter options):

- Use the **SA Filter** to define the types of service agreements whose financial transactions appear in the grid. The following options are available:
 - **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.

- **Geographic Type.** Use this option to restrict service agreements to those linked to service points associated with a given **Geo Type** and **Value**.
- **SA ID.** Use this option to see a specific **Service Agreement**.
- **SA Type.** Use this option to restrict service agreements to those with a given **CIS Division** and **SA Type**.
- Use **Status Filter** to restrict the service agreements based on whether the sum of the debits and credits they contribute to the match event. The following options are available:
 - **All.** Use this option if you do not wish to restrict service agreements based on this status.
 - **Balanced.** This option shows only service agreements where the sum of debit and credits nets to zero on this match event.
 - **Unbalanced.** This option shows only service agreements where the sum of debit and credits do not net to zero on this match event.

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. The following information appears in the grid:

- **SA Information.** This column contains a summary of important information about the service agreement.
- **Difference.** This column contains the difference between the **Matched Debits** and **Match Credits**. If this is non-zero, the value appears in red.
- **Matched Debits.** This column contains the sum of debit financial transactions that this service agreement contributes to this match event.
- **Matched Credits.** This column contains the sum of credit financial transactions that this service agreement contributes to this match event.
- **Premise Information.** This column contains a summary of the premise (if any) associated with the financial transaction's service agreement.

How To Perform Common Match Event Functions

Contents

- [How To Find The Match Event Associated With A Financial Transaction](#)
- [How To Dispute An Item](#)
- [How To Match A Small Mismatch](#)

How To Find The Match Event Associated With A Financial Transaction

If you need to find the match event on which a financial transaction (FT) was matched, display the FT in question on [Financial Transaction - Main](#) and then use the "go to" button adjacent to the **Match Event** to drill to the match event.

Note. The easiest way to display a financial transaction is to find its corresponding [bill](#), [payment](#) or [adjustment](#) and then drill down on the desired transaction.

How To Dispute An Item

Refer to [Disputing Items](#) for background information about disputes.

If a customer wants to dispute an item:

- Create a match event for the account.
- Turn the match event's dispute switch on.
- Link the disputed financial transaction to the match event.
- Describe in the match event's comments the reason for the dispute.

How To Match A Small Mismatch

Assume the following scenario arises:

- A bill is produced for \$2000
- The customer pays \$1993
- An unbalanced match event will result because the customer didn't pay exactly what is owed

If you want to match this payment to the bill (and leave \$7 for the next bill), do the following:

- Create a transfer adjustment of \$7 where the transfer from / to service agreement is the same. This results in a debit financial transaction (FT) of \$7 and a credit FT of \$7.
- Create a match event (or update the unbalanced match event) where the matched FTs are:
 - The \$1993 payment
 - The credit side of the transfer adjustment (\$7)
 - And the \$2000 bill
- Then, if the customer pays their next bill in full, the \$7 debit (associated with the transfer adjustment) will be swept onto it.

Automating small mismatches. The algorithm responsible for matching a payment to a specific bill can have a tolerance amount defined on it. If the payment is within the tolerance limit, this algorithm will do the above for you. In other words, you don't have to manually do the above if you populate the tolerance limit appropriately on this algorithm. Refer to [DSOV BILL-ID](#) for more information about this algorithm.

Deposits

In this section, we describe how to manage your deposits.

Contents

[The Big Picture Of Deposits](#)
[Deposit Review](#)

The Big Picture Of Deposits

The topics in this section provide background information about a variety of deposit issues.

We strongly recommend familiarizing yourself with the topics described in [The Financial Big Picture](#) to fully appreciate how deposits fit into the system's financial architecture.

Contents

[Cash Deposits](#)
[Refunding Deposits](#)
[Partial Refunds](#)
[Total Amount To Bill](#)
[Current Balance versus Payoff Balance](#)
[Multiple Deposits Linked To A Single Account](#)
[Deposits and Credit & Collections](#)
[Deposit Seizures When Other SAs Are Stopped](#)
[Non-Cash Deposits](#)
[3rd Party Deposits](#)
[Deposit Class Controls Everything](#)
[Deposit Background Processes](#)

Cash Deposits

Before you can bill a customer for a deposit, there must be a deposit service agreement (SA). Deposit SA's will be created by a CSR using [Start/Stop Service](#) (the same page used to create all other types of service agreements).

Deposit SA's behave just like any other service agreement in that:

- A deposit SA must reference an SA type. Refer to [Service Agreement Type Controls Everything](#) for information about how SA type controls a service agreement's behavior.
- Bill segments will be produced to bill the customer for a deposit. Note, bill segments associated with deposit SA's will appear on the same bill as other bill segments related to its account (i.e., a bill could contain a combination of utility, deposit, and non-utility bill segments).
- When a customer makes a payment, it will be distributed amongst its account's SAs based on each SA's payment distribution priority. This means a single payment could relieve receivable accounts (associated with utility service agreements) and increase a payable account (associated with the deposit service agreement). Refer to [Distributing A Payment Amongst An Account's Service Agreements](#) for more information.

- You can use a transfer adjustment to relieve debt on some other type of service agreement (by transferring all or part of the deposit's credit balance to another service agreement).
- You can use an A/P adjustment to cause a check to be created if you need to manually refund a deposit.
- When you no longer require the deposit, a CSR will stop the deposit SA using [Start/Stop Service](#) (the same page used to stop all other types of service agreements). Refer to [Refunding Deposits](#) for information about how the refund actually takes place.

Refer to [Deposit Class Controls Everything](#) for information about deposit interest, the recommended deposit amount algorithm, and how deposits are refunded.

An account can have many deposits. You can view all deposits (both cash and non-cash) linked to an account on [Account – Deposits](#).

Refunding Deposits

To refund a deposit, you simply need to change the deposit SA's state to **pending stop**. The system does everything else (i.e., calculates interest and refunds the deposit to the customer).

There are three ways in which a deposit SA's state can become **pending stop**:

- The Deposit Refund background process (referred to by the batch control ID **DEPRFND**) make a deposit service agreement **pending stop** when it detects that the customer meets the automatic refund criteria. Refer to [Defines The Conditions That Cause The System To Automatically Refund A Deposit](#) for a description of how to define the conditions that control automatic refund.
- An operator, at their discretion, may refund a deposit by stopping the deposit SA using [Start/Stop Service](#) (the same page used to stop all other types of service agreements). This page simply changes the deposit SA's state to **pending stop**.
- The system changes the state of a deposit service agreement to **pending stop** when it recognizes that all other SA's of its deposit class have also been **stopped**. This may be necessary, for example, when a customer's utility service agreements have been severed for non-payment. Refer to [Deposit Seizures When Other SAs Are Stopped](#) for more information.

The system changes the state of a **pending stop** SA to **stopped** when the process responsible for [Finalizing Pending Stops](#) next runs.

When a deposit SA becomes **stopped**, the system calculates interest through the stop date. Refer to [Controls Interest Calculation](#) for a description of how interest is calculated.

Note. If you can't wait for this process to run, simply display the deposit SA on [Service Agreement – Main Information](#) and press the stop button.

When the **stopped** deposit SA is final billed, the system produces a final bill segment that shows the deposit amount being returned to the customer. When the bill segment's bill is completed, the deposit is refunded to the customer as per the deposit refund algorithm on the deposit's deposit class. Refer to [Controls How Deposits Are Refunded To A Customer](#) for a description of how a deposit is refunded.

Warning! None of the base package deposit refund algorithms create an A/P adjustment when the deposit service agreement is final billed. Rather, they depend on the [write-off preprocessing](#) to actually refund the deposit to the customer.

Partial Refunds

You can perform a partial refund by creating an adjustment for the deposit SA. If you want to refund the adjustment with a check, this adjustment should be an A/P adjustment (i.e., one that is interfaced to A/P). If you want to refund the adjustment by offsetting some other service agreement's debt, create a transfer adjustment (transferring from the deposit SA to the SA to be relieved).

After creating the adjustment, you'll find that the deposit SA has a positive current amount due (because you debited the deposit SA to refund the credit amount). You should create another adjustment to return the current amount of the deposit to zero. It's important to use an adjustment type that only affects current amount when you do this.

After getting the deposit's current amount back to an expected amount, you will have to change the deposit SA's Total Amount To Bill to reflect the amount of deposit you want to hold for the customer. So, for example, if you were holding a \$500 deposit and you want to refund \$100, you should change the deposit SA's Total Amount To Bill to be \$400.

Total Amount To Bill

Most deposit service agreements contain a "Total Amount To Bill" field. The label for this field is defined on the SA type (on the Billing tab). For deposit SAs, it should be labeled something like **Cash Deposit Amount**. This field should reflect the deposit amount you WANT to hold on this service agreement. Please be aware of the following in respect of this field:

- If the payoff balance (i.e., the current deposit held) is less than the Total Amount To Bill, the system will generate a bill segment to charge the customer for the deposit.
 - If the deposit service agreement uses a recurring charge (i.e., you bill the total amount in installments), the amount billed to the customer will not exceed the recurring charge amount (unless the bill segment is prorated).
 - If the deposit service agreement does not use a recurring charge, the amount billed will equal the difference between Total Amount To Bill and the service agreement's payoff balance.
- If you need to bill an incremental deposit on an existing deposit service agreement, simply change the deposit service agreement's Total Amount To Bill.
- If the payoff balance (i.e., the current deposit held) is greater than the Total Amount To Bill, the system will NOT refund the excess deposit.

SA Type and Total Amount to Bill. A deposit service agreement's SA type controls if the deposit service agreement uses Total Amount to Bill. Typically, the only SA types that indicate this field can be used are loan and deposit service agreements.

Current Balance versus Payoff Balance

Warning! If you do not understand the difference between payoff balance and current balance, refer to [Current Amount versus Payoff Amount](#).

The current balance on a deposit SA contains the amount of the deposit that has been billed, but not paid. The payoff balance on a deposit SA contains the amount of deposit being held. The payoff balance is a negative number because your company owes this money to the customer.

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

Note. It's important to be aware that everything that is shown in this table is controlled by how you set up the deposit SA type's bill segment type, payment segment type, and adjustment types. Refer to [Service Agreement Type Controls Everything](#) for how to do this.

Event	GL Accounting	Arrearage Rule	Effect On Payoff Amt	Effect On Current Amt	Payoff Balance	Current Balance
Deposit billed	N/A – the GL is not affected when a deposit is billed	\$100 starts aging	0	+100	0	100
Payment received	Cash 100 Deposit Payable <100>	\$100 relieved accordingly	-100	-100	-100	0
Interest calculated	Interest Exp 5 Deposit Payable <5>	N/A	-5	0	-105	0

The following points describe the events in the above table:

- **Deposit billed.** In this example, the customer is billed for \$100 deposit.
 - The customer really thinks they owe the billed amount, \$100. Therefore, current amount is affected. However, if the customer was to cash out, they wouldn't owe your organization anything, therefore payoff amount is not affected.
 - Notice that the GL is not affected when the deposit is billed. This is because most organizations do not show a receivable for billed deposits (as it's not a true receivable).
 - Because current amount changed by \$100, arrearage history is affected accordingly.
- **Payment received.** With any luck, the client will pay the \$100 that was billed.
 - The payment has a normal affect on the GL (debit cash, credit deposit payable). The deposit payable GL account is defined as the deposit SA type's Distribution Code.
 - The amount the customer thinks they owe decreases by \$100, therefore current amount is affected by the payment amount. And, if the customer was to cash out, your organization would owe the customer \$100, therefore payoff amount is affected by the payment amount.

- Because current amount changed by \$100, arrearage history is affected accordingly.
- **Interest calculated.** In this example, the system calculates interest of \$5.
 - The interest is posted to the GL (the interest expense distribution code is defined on the respective adjustment type).
 - The interest amount didn't affect how much the customer thinks is due them. Therefore current amount is unaffected. However, if the customer was to cash out, your organization would owe them \$100 + \$5 (the interest) therefore payoff amount is affected by \$5.
- Because current amount is not changed, arrearage history is not affected.

Multiple Deposits Linked To A Single Account

The system has been designed to allow your customers to have multiple deposits. This is necessary when a deposit is restricted to a specific type of debt. For example, if separate deposits are held for regulated and unregulated debt (and a customer could hold a combination of regulated and unregulated debt), you'd need one deposit for regulated debt and another for unregulated debt.

Deposit class. A deposit's deposit class controls the service agreements that are covered by a given deposit. A deposit's deposit class is defined on its SA type. Refer to [Designing Your Deposit Classes](#) for more information.

It's important to be aware that if your company has multiple classes of deposits, the system will recommend and refund deposits for each individual deposit class. For example, if you have both regulated and unregulated deposit classes, the system will recommend separate deposits for each class. And at refund time, a deposit will be distributed to only those service agreements associated with its deposit class (i.e., a regulated deposit will not be used to satisfy unregulated debt).

Deposits and Credit & Collections

The [account debt monitor](#) monitors deposit service agreements just as it monitors every other service agreement for overdue debt. If the amount of debt on the deposit SA violates your collection criteria, a collection process will start.

Refer to [The Lifecycle Of A Collection Process And Its Events](#) for more information.

It's important to be aware that deposit debt can be treated differently from other types of debt linked to an account by creating a specific debt class for the deposit SA type.

Refer to [Designing Your Collection Procedures](#) for more information about how to have different collection criteria for different debt classes.

The type of severance process associated with a deposit SA is probably rather simple – you probably will want a To Do entry generated to advise an operator that a customer hasn't paid their deposit.

Refer to [The Lifecycle Of A Severance Process And Its Events](#) for more information.

Please see [Deposit Seizures When Other SA's Are Stopped](#) for a description of how the system seizes an active deposit service agreement when normal (i.e., non-cash deposit) service agreements are stopped.

After a deposit SA is **stopped**, it will be “final billed” the next time the account is billed. Refer to [Refunding Deposits](#) for more information about how deposit refunds are “final billed”. If a credit balance remains on the deposit SA after it is final billed, the [Write-Off Monitor](#) will process the deposit SA just like it processes all unpaid and final billed service agreements. If you set up the deposit SA's [write-off controls](#) properly (i.e., you plug in the appropriate refund and write-down algorithms), the write-off monitor will refund / write-down the credit balance (thus causing the deposit SA to close).

Deposit Seizures When Other SAs Are Stopped

Consider the situation when a severance process cuts a service due to non-payment. At some point, if the customer doesn't pay, you'll want to seize the deposit and apply it to the overdue debt. To implement this, the system simply stops the deposit SA when it stops the last SA in its deposit class (refer to [Finalizing Pending Stops](#) for more information about stopping service agreements). Then, when the account is next billed, the standard deposit refund process will offset outstanding debt before it is refunded to a customer.

Refer to [Refunding Deposits](#) for a complete description of how the system refunds a stopped deposit SA's balance to other service agreements in its deposit class.

Deposit class refund method is the key. It's important to stress that the only way the system will use a deposit to offset overdue debt is if the deposit's deposit class has a refund method that does this. If you've set up a deposit class' refund method to cut a check for the entire deposit amount, the system will not offset outstanding debt. Refer to [Deposit Class – Refund Method](#) for more information.

Non-Cash Deposits

When an account is required to post a deposit, they can remit cash or some other type of surety (e.g., letters of credit, surety bonds, 3rd party deposits). When cash is remitted, a deposit service agreement is used. When some other type of surety is used, a *non-cash deposit* must be created.

Non-cash deposits are held in respect of an account and an account may have an unlimited number of non-cash deposits (note: cash deposits are held in respect of a deposit service agreement that is linked to remitting account). Refer to [Account – Deposits](#) for more information.

Each non-cash deposit must reference a non-cash deposit type. Besides defining the type of surety, the non-cash deposit type also controls:

- Whether such types of non-cash deposits can be highlighted when they are due to expire via the creation of a To Do entry (refer to [TD-NCDEX](#) for more information about this To Do entry).
- Whether a reference to a cash-deposit (i.e., service agreement) must be specified. Refer to [3rd Party Deposits](#) for more information about how a cash deposit can be referenced by another party.

Refer to [Setting Up Non-Cash Deposit Types](#) for more information.

In addition to non-cash deposit type, each non-cash deposit must reference a deposit class. Why? Because the system amalgamates cash and non-cash deposits when it determines if an account is holding an adequate deposit.

Refer to [What Do Deposit Classes Do?](#) for more information.

If you need to use a non-cash deposit to satisfy an account's debt, you must "cash in" the non-cash deposit. When the cash arrives, create a payment and apply it to the customer's outstanding service agreements.

3rd Party Deposits

A 3rd party deposit arises when a 3rd party remits cash to cover the deposit needs of one or more accounts. Both cash and non-cash deposits must be created to record a 3rd party deposit. The following example will explain how to do this.

Assume the Salvation Army remits \$5,000 on behalf of 50 customers (where each customer's account is allocated \$100). In this situation, you'd create the following information in the system:

- The Salvation Army must have an account with a deposit service agreement.
- When the Salvation Army remits the funds:

Warning! If you do not understand the difference between payoff balance and current balance, refer to [Current Amount versus Payoff Amount](#).

- Create an adjustment to "bill" the deposit service agreement (causing the deposit SA's current balance to be \$5,000 and the payoff balance to be 0).
- Add a payment for the \$5,000 against the Salvation Army's account. This payment will cause the deposit SA's current balance to be 0 and the payoff balance to be -\$5,000. Note, if you don't mind the Salvation Army's current balance to be -\$5,000 after the payment is made, you wouldn't have to create the adjustment to "bill" the deposit.
- Create a non-cash deposit for each of the 50 accounts being covered by the cash deposit. On each non-cash deposit, define the appropriate non-cash deposit type (e.g., 3rd party deposit) and amount \$100 each.
- Interest will be applied to the Salvation Army's deposit service agreement as per the interest algorithm on the deposit SA's deposit class.
- If you need to use the Salvation Army's payment to payoff overdue debt, you will use a transfer adjustment(s) to transfer from the deposit service agreement to the respective overdue service agreement(s).

Important! Be aware that the system will allow the sum of 3rd party deposits to exceed the amount of the cash deposit (in our previous example, the system would allow you to create \$6,000 worth of 3rd party non-cash deposits even though only \$5,000 of cash was remitted).

Deposit Class Controls Everything

A deposit service agreement's SA type references a deposit class. The topics in this section describe how deposit class controls the behavior of a deposit SA.

Refer to [Setting Up Deposit Classes](#) for a complete description of the business rules governed by a deposit class.

Contents

- [Controls Interest Calculation](#)
- [Controls The Recommended Deposit Amount](#)
- [Defines The Conditions That Cause The System To Recommend A New or Additional Deposit](#)
- [Defines Automatic Refund Conditions](#)
- [Controls How Deposits Are Refunded To A Customer](#)

Controls Interest Calculation

Interest is applied to a deposit SA when:

- A deposit SA is **stopped**. Refer to [Refunding Deposits](#) for a description of how a deposit SA can be stopped.
- The **Apply Interest** background process (referred to by the batch control ID **DEPINTRF**) determined that sufficient time has passed since interest was last calculated. The amount of time between interest calculations is defined on the deposit class.


A deposit SA's deposit class controls the following interest calculation functions:

- The frequency of interest calculation.
- The algorithm used to calculate the amount of interest (note, the interest rate is defined using a bill factor plugged in on this algorithm).
- The method used to refund the interest.

Adjustments are used to apply interest. Interest is applied to a deposit SA using an adjustment (the adjustment type is also defined on the deposit class).

Controls The Recommended Deposit Amount

The following events cause the calculation of the recommended deposit amount:

- The **Deposit Review** background process (referred to by the batch control ID **DEPRVW**) compares an account's existing deposit (if any) to the recommended amount.
- A user requests a deposit calculation for a new deposit service agreement by pressing the  button on the [Start Confirmation](#).

A deposit service agreement's deposit class controls the algorithm used to derive the suggested deposit amount.

Defines The Conditions That Cause The System To Recommend A New or Additional Deposit

The Deposit Review background process (referred to by the batch control ID **DEPRVW**) will recommend an additional deposit be billed if the customer doesn't satisfy the definition of a *good customer*.

It's important to be aware that if your company has multiple classes of deposits, the system will recommend deposits for each individual deposit class used by an account. For example, if an account has both regulated and unregulated deposit classes, the system will recommend separate deposits for each class.

When the Deposit Review process analyzes an account's debt, it uses the following deposit class attributes:

- The definition of a good customer (the system only recommends deposits for bad customers).
- The tolerance percent that must be exceeded before the system will recommend an additional deposit for a customer. This tolerance percent prevents the recommendation of small amounts.
- The method used to derive the recommended deposit amount.

Assuming the [C1-CR-UP-DRR](#) review method algorithm is specified on the deposit class, the system does not automatically produce bills when an additional deposit is needed. Rather, the system's recommendations appear on the [Deposit Review](#) page. To implement a recommendation, an operator should change the deposit SA's [Total Amount To Bill](#) or add a new deposit SA.

Non-cash deposits are included. It's important to be aware that the system compares the recommended deposit amount against the sum of deposits on hand for a given deposit class. When amalgamating the total deposit on hand, the system includes both cash (i.e., deposit SA's payoff balance) and non-cash deposits.

Defines Automatic Refund Conditions

The Deposit Refund background process (referred to by the batch control ID **DEPRFND**) will refund a deposit to a customer when conditions defined on the deposit SA's deposit class are met. For example, a deposit's deposit class may indicate the system should automatically refund a deposit after the deposit has been held for 6 months and the customer is a good customer (the definition of a "good customer" is also on the deposit class).

Controls How Deposits Are Refunded To A Customer

The method used to refund a deposit to a customer is defined on the deposit SA's deposit class. For example, a deposit's deposit class may indicate the system will first apply the deposit to outstanding debt first and, if funds remain, cut a check.

It's important to be aware that if your company refunds deposits by first offsetting outstanding debt, the system will only offset debt within an individual deposit class. For example, if a customer has two deposits – one for regulated debt, the other for unregulated debt – the regulated deposit will only be used to offset regulated debt.

Deposit Background Processes

The topics in this section describe the background processes that automate deposit processing.

Contents

[Deposit Interest](#)

[Deposit Refund](#)

[Review Deposits](#)

Deposit Interest

The Deposit Interest background process (referred to by the batch control ID **DEPINTRF**) examines all cash deposit service agreements (i.e., service agreements with a SA type with a special role of **Cash Deposit**). If enough time has passed since interest was last calculated, interest will be calculated using the service agreement's deposit class' Interest Refund Algorithm. Note: interest will be automatically calculated every X months where X is defined in the deposit class' Months Between Interest Refund.

Deposit Refund

The Deposit Refund background process (referred to by the batch control ID **DEPRFND**) examines all cash deposit service agreements (i.e., service agreements with a SA type with a special role of **Cash Deposit**). If the account meets the service agreement's deposit class' Refund Criteria Algorithm, the system changes the service agreement's state to **Pending Stop**. Refer to [Refunding Deposits](#) for a description of what happens next.

Review Deposits

The Review Deposits background process (referred to by the batch control ID **DEPRVW**) examines all accounts with service agreements that are governed by a deposit class (i.e., service agreements with an SA type that references a deposit class). If the account fails the deposit class' Good Customer Algorithm, the system calculates the recommended deposit amount (using the deposit class' Deposit Recommendation Algorithm). If the recommended amount exceeds the amount of deposit currently requested, the system will request an additional deposit. The deposit class' **Review Method Algorithm** is used to determine what action to take if the system requests an additional deposit. The system's recommendations appear on the [Deposit Review](#) page.

You may optionally provide a Deposit Class as input to restrict the review to account whose service agreements reference that deposit class.

Preventing small recommendations. The system uses the deposit class's Review Tolerance Percentage to prevent the recommendation of small deposits by the Deposit Review background process. For example, if this field contains 10(%), the system would only recommend an additional deposit if the existing requested deposit amount is less than 90% of the recommended amount.

Multiple deposits could be recommended for an account. Because an account can have service agreements that belong to multiple deposit classes, it is possible for a multiple recommendations to be generated for an account.

Deposit Review

The Deposit Review page shows all accounts for which an additional deposit is recommended.

Note. The [Review Deposits](#) background process inserts rows on this query.


Multiple deposits could be recommended for an account. Because an account can have service agreements that belong to multiple deposit classes, it is possible for a multiple recommendations to be generated for an account.

Open this page using **Financial Query, Deposit Review**.

Description of Page

Use **Account ID** and / or **Deposit Class** to filter the rows. Each row displays an account / deposit class that holds an inadequate deposit.

Rows are also displayed for errors. Please be aware that if the Recommend Additional Deposit background process encounters an error when it attempts to calculate an account's recommended deposit, these errors also appear as rows in this query.

Press the  button to add a new deposit SA for the customer. Pressing this button causes [Start/Stop Service](#) to open. When this page opens, you should create a new deposit service agreement for the customer (by referencing the appropriate division / SA Type).

If the customer has one or more deposit service agreements, the drill button will be enabled. Pressing it will cause the [Service Agreement – Main Information](#) page to open. When this page opens, you can change Total Amount To Bill to reflect the total amount of deposit you want to hold. When billing next runs, it will bill the customer for the marginal difference between the current deposit held and the deposit amount to bill.

Refer to [Defines The Conditions That Cause The System To Recommend A New or Additional Deposit](#) for how rows are added to this query.

Interval Billing

The interval billing functional area is responsible for managing the following:

- Collecting interval data. Any time increment can be supported. For example, electrical meter read data might be recorded in 10 minute, 15 minute, 30 minute, ... intervals; whereas daily gas consumption may have intervals that span 24 hours.
- Maintaining interval prices. Again, any time increment can be supported.
- Deriving billable interval consumption from multiple interval consumption sources. For example, actual interval consumption can be compared against a customer-specific “maximum demand” profile to derive an “excess demand” profile.
- Deriving time-of-use consumption by applying time-of-use maps to a customer’s interval consumption.

This chapter describes the above points in detail.

Note. The transactions described in this document are available only if the [interval billing](#) modules are not [turned off](#).

Contents

[Interval Pricing Background Topics](#)
[The Big Picture of Raw Data Collection and Aggregation](#)
[Time of Use Mapping Background Topics](#)
[Contract Option Background Topics](#)
[Maintaining Interval Data](#)
[Interval Billing Examples](#)

Interval Pricing Background Topics

The topics in this section provide background information about a variety of interval billing issues.

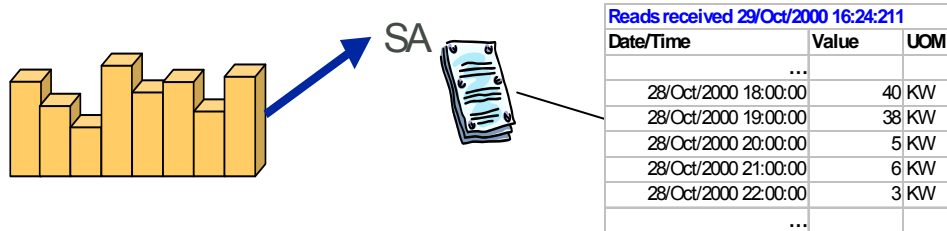
Contents

[Billable Interval Quantities for a Service Agreement](#)
[Interval Quantities are Stored in Data Sets](#)
[Fixing Errors with Complete Data Sets](#)
[Algorithms Find the Most Up To Date Data](#)
[Time Issues](#)
[Interval Data Serves a Role for a Service Agreement](#)
[Common Profiles vs. SA Owned Profiles](#)
[Flexibility of SA / Profile Link](#)
[The Link between Profile and SA is Effective Dated](#)
[Creation of Profile Data through Data Derivation](#)
[Validation of Profile Data](#)
[Auditing Your Interval Pricing Bill Lines](#)

Billable Interval Quantities for a Service Agreement

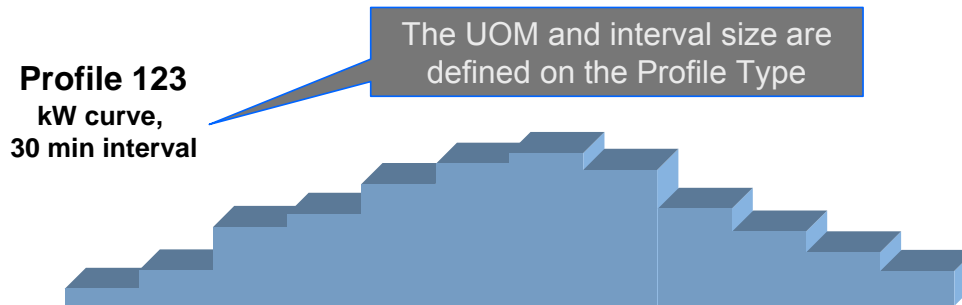
Billable interval quantities are linked to the service agreement. These quantities are linked through an Interval Profile. The interval data that is received over time is linked to the service agreement in one of the following ways:

- It is uploaded from an external source. Refer to [Upload Interval Data](#) for more information.



- It may be created in the system through data derivation algorithms. Refer to [Creation of Profile Data through Data Derivation](#) for more information.

All the data received throughout time could be plotted as a curve. For interval quantities, the curve is referred to as an Interval Profile.



The profile references a Profile Type that defines the physical attributes of the data curve. It defines the unit of measure, the interval size and may also contain data derivation algorithms, which actually populate the profile with data based on other information. Refer to [Designing Interval Profile Types](#) and [Creation of Profile Data through Data Derivation](#) for more information.

Interval Quantities are Stored in Data Sets

Interval data in the system may have many different origins.

- It may be contract-related data that is set up at the beginning of the contract or contract year.
- It may be data that is periodically interfaced from an external source.
- It may be data that is derived and created through a system algorithm.

Regardless of the origin of the data, all data has similar characteristics:

- Data is received by the system or created by the system periodically. The system stores 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.

The system accepts the new batches of data for a given profile and stores the data grouped together by a data set.

Prices for Bill Factor “Market Price” and Characteristic Value of “Market A”

Market Prices for Market A	
Prices received 29/Oct/2000 16:38:228	
Date/Time	Value
...	
30/Oct/2000 18:00:00	\$0.051
30/Oct/2000 19:00:00	\$0.042
30/Oct/2000 20:00:00	\$0.039
30/Oct/2000 21:00:00	\$0.038
30/Oct/2000 22:00:00	\$0.039
...	

← Data Set

← Data

Market Prices for Market A	
Prices received 1/Nov/2000 08:37:109	
Date/Time	Value
...	
30/Oct/2000 19:00:00	\$0.058
30/Oct/2000 21:00:00	\$0.043
31/Oct/2000 02:00:00	\$0.021
31/Oct/2000 03:00:00	\$0.022
31/Oct/2000 04:00:00	\$0.021
31/Oct/2000 05:00:00	\$0.019
...	

← Data Set

← Data

There are two effective dates referenced in the interval pricing information: **Set Date/Time** and **Interval Value Date/Time**. Think of the **Set Date/Time** as the effective date of the entire set of interval data. Think of **Interval Date/Time** as the effective date of the associated piece of data.

Refer to [Algorithms Find the Most Up To Date Data](#) for information about processing this data.

Fixing Errors with Complete Data Sets

If a data set is complete, but the values are incorrect, you are not able to cancel the data set. In order to reverse the effect of a completed data set, you must create a new completed data set with corrected data.

Original data received		
Reads received 29/Oct/2000 16:24		
Date/Time	Value	UOM
...		
28/Oct/2000 18:00:00	40	KW
28/Oct/2000 19:00:00	38	KW
28/Oct/2000 20:00:00	5	KW

New data received to override previous data		
Reads received 30/Oct/2000 12:39		
Date/Time	Value	UOM
...		
28/Oct/2000 18:00:00	42	KW
28/Oct/2000 19:00:00	40	KW
28/Oct/2000 20:00:00	6	KW

New Data Set created 31/Oct/2000		
If the data received on 30/Oct is incorrect, then a new data set with the 29/Oct data should be created		
Date/Time	Value	UOM
...		
28/Oct/2000 18:00:00	40	KW
28/Oct/2000 19:00:00	38	KW
28/Oct/2000 20:00:00	5	KW

← This is called 'reverting' the data

What happens if no previous values exist for the incorrect data? There are no values available to replace the incorrect values. You essentially want to indicate to the system that no data exists for this interval. To do this, you create a data set with a special data set type of **No Data**.

New data received,
adds new intervals

Reads received 30/Oct/2000 08:24

Date/Time	Value	UOM
...		
28/Oct/2000 21:00:00	8	KW
28/Oct/2000 22:00:00	7	KW
28/Oct/2000 23:00:00	8	KW

To "cancel" this data, create
a "No Data" data set with a
more recent date

New Data Set created 31/Oct/2000

Data Set Type = "No Data"		
Date/Time	Value	UOM
...		
28/Oct/2000 21:00:00	0	KW
28/Oct/2000 22:00:00	0	KW
28/Oct/2000 23:00:00	0	KW

The system provides a Revert button on the appropriate interval data pages to help the user to reverse the effect of a data set. When reverting, the system creates a new data set and populates the intervals with the most recent values for those intervals, ignoring the values in the data set being reverted. If there are no other values for any of the interval, then a "No Data" data set is created for these intervals.

Using Revert, the system creates
one data set for intervals with
previous data ...

Original data received

Reads received 29/Oct/2000 16:24

Date/Time	Value	UOM
...		
28/Oct/2000 18:00:00	40	KW
28/Oct/2000 19:00:00	38	KW
28/Oct/2000 20:00:00	5	KW

New data received that
overrides data AND adds
new intervals

Reads received 30/Oct/2000 12:39

Date/Time	Value	UOM
...		
28/Oct/2000 18:00:00	42	KW
28/Oct/2000 19:00:00	40	KW
28/Oct/2000 20:00:00	6	KW
28/Oct/2000 21:00:00	9	KW
28/Oct/2000 22:00:00	8	KW
28/Oct/2000 23:00:00	9	KW

New Data Set created 31/Oct/2000

Date/Time	Value	UOM
...		
28/Oct/2000 18:00:00	40	KW
28/Oct/2000 19:00:00	38	KW
28/Oct/2000 20:00:00	5	KW

New Data Set created 31/Oct/2000 Data Set Type = "No Data"

Date/Time	Value	UOM
...		
28/Oct/2000 21:00:00	0	KW
28/Oct/2000 22:00:00	0	KW
28/Oct/2000 23:00:00	0	KW

... and a data set indicating
"No Data" for intervals with
no previous data

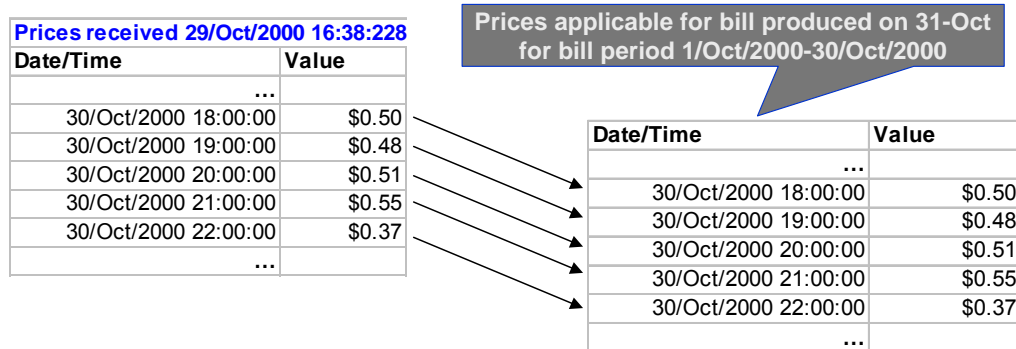
Refer to [Interval Data Maintenance](#), [TOU Data Maintenance](#) and [Interval Register Data Maintenance](#) for more information about setting the data set type flag and using the **Revert** button.

Algorithms Find the Most Up To Date Data

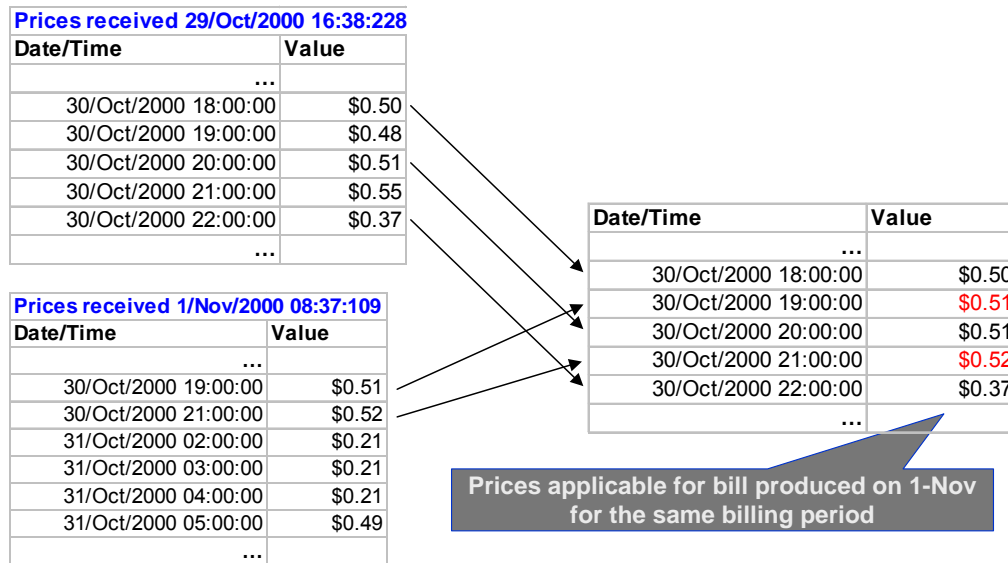
Any algorithm that must retrieve interval data, whether it is bill factor interval values or time of use map data or interval profile data, needs to ensure that it accesses the most up to date data. This includes algorithms used by the rates to process interval data and any derivation algorithm.

Refer to [Interval Quantities are Stored in Data Sets](#) for information about how interval data is stored.

The following diagram illustrates data retrieved by rates to use for billing a customer on a given date.



The following diagram illustrates what happens when data is received by the system that is a correction of previously received data. A bill produced after this data is received uses the most up to date values.



Whenever a collection of interval data needs to be accessed, the system checks the following:

- As of what date? This enables the system to find the data sets that existed in the system on that date. It looks for the appropriate data within these data sets.
- What is the time period? This enables the system to get the correct collection of interval data.

If you want to verify the data used to produce a bill,

- The “as of what date” is the bill segment creation date
- The bill segment start and end dates define the time period

Time Issues

Contents

Start and End Times for Billing Time Zone and Time Changes

Start and End Times for Billing

As you know, there is logic in billing to determine the start date and end date for a bill segment. Refer to [Ways to Control The End Date Of A Bill](#) for more information. When billing for a customer with interval data, the system also needs to know the time.

The time used by billing, referred to as the cutoff time, is stored on the service agreement. There is also a control on the service agreement called Start Day Option that determines which day to use for the start time. Billing algorithms use the billing date, the cutoff time, and the start day option to determine the correct interval data to process.

Contents

[Billing Time Examples](#)
[First and Last Bill Segment Considerations](#)
[Defaulting Service Agreement Time Options](#)
[Changing Service Agreement Time Options](#)

Billing Time Examples

Let's look at some examples.

Contents

[Current Day Option](#)
[Previous Day Option](#)
[First Bill Segment](#)

Current Day Option

- Assume that the [consumption period](#) is from the 1st of October through the 29th of October.
- Assume the cutoff time on the service agreement is 2 a.m. and the start day option is Current Day.

Date/Time	Data
...	
01/Oct/2000 01:00:00	\$0.51
01/Oct/2000 02:00:00	\$0.52
01/Oct/2000 03:00:00	\$0.21
01/Oct/2000 04:00:00	\$0.21
...	
30/Oct/2000 00:00:00	\$0.51
30/Oct/2000 01:00:00	\$0.52
30/Oct/2000 02:00:00	\$0.21
30/Oct/2000 03:00:00	\$0.21
30/Oct/2000 04:00:00	\$0.21
...	

The first interval used is the one **AFTER** the cutoff time on the bill start date

The last interval used is the one that includes the cutoff time on the day **AFTER** the bill end date

To determine the starting interval, the system finds the first interval after the cutoff time on the billing start date. The system uses all the intervals up to and including the cutoff time for the DAY AFTER the billing end date.

Previous Day Option

- Assume that the [consumption period](#) is from the 1st of October through the 29th of October.

- Assume the cutoff time on the service agreement is 10 p.m. and the start day option is Previous Day.

Date/Time	Data
...	
30/Sep/2000 22:00:00	\$0.51
30/Sep/2000 23:00:00	\$0.52
01/Oct/2000 00:00:00	\$0.21
01/Oct/2000 01:00:00	\$0.21
...	
29/Oct/2000 21:00:00	\$0.51
29/Oct/2000 22:00:00	\$0.52
29/Oct/2000 23:00:00	\$0.21
30/Oct/2000 00:00:00	\$0.21
30/Oct/2000 01:00:00	\$0.21
...	

The first interval used is the one **AFTER** the cutoff time on the day **BEFORE** the bill start date

The last interval that is the one which includes the cutoff time on the bill end date

To determine the starting interval, the system finds the first interval after the cutoff time on the DAY BEFORE the billing start date. The system uses all the intervals up to and including the cutoff time for the billing end date.

Refer to [Bill Period and Seasonal Time Shifts](#) for more information about the effect of time shifting on determining the bill period.

First Bill Segment

When billing calculates the consumption period for the very first bill segment for your service agreement, it uses a flag on your SA type to correctly calculate the [initial consumption period](#). For your interval billing service agreements, you typically configure your SA types to indicate that the start of the consumption period for the first bill should include the SA start date unless this is a 'back-to-back' situation (i.e., a previous customer was already billed for that day as its end date).

Let's look at an example. Imagine your SA type is configured to **Add 1 Day for Back-to-back** and this is not a back-to-back situation. Also imagine that your service agreement has the following defined:

- Start Date: **April 1, 2003**
- Start Day Option: **Current**
- Cutoff Time: **2am**

The start period for the consumption period would be set to **April 1, 2003; 2am**.

Imagine the same SA exists, but this is a back-to-back situation. In other words, the previous customer ended service on April 1, 2003. Assuming the previous customer has the same start day option and cutoff time, they are billed through **April 2, 2003; 2am**. Refer to [Current Day Option](#) for more information. In this case, billing calculates the start period for the consumption period as **April 2, 2003; 2am**.

First and Last Bill Segment Considerations

Billing provides the consumption period used for defining the start and ending intervals by the interval pricing and TOU pricing algorithms.

However, for the very first bill segment for the customer, perhaps your business practice dictates that the first interval is at some time other than the cutoff time. For example, rather than billing from 2 a.m. on the first day, perhaps the meter is only installed at 3 p.m. and that is when billing should begin. Similarly, for the final interval on the last bill segment for your service agreement, you may require billing to end when the meter is removed at 4 p.m., rather than through 2 a.m. on the day after the end date.

To accomplish this functionality, follow these guidelines:

- When starting or stopping a service agreement, navigate to the [service agreement interval info](#) tab to define the appropriate starting or ending interval for each interval profile or TOU map linked to the service agreement using the start and end date/times in the profile or map collection.

Note. If the start/end intervals are defined based on the date/time a meter was installed or removed, the recommendation is to design a mechanism for updating the SA/profile and SA/TOU map date/times via an interface. For example, perhaps a field activity completion algorithm could update the profile and/or map date/times according to when the meter was installed. Or perhaps a notification & workflow process is designed to update this information.

- Design your interval pricing and/or TOU pricing algorithms to override the consumption period information passed in for the first interval of the first bill segment or for the last interval for the final bill segment. These algorithms receive the service agreement effective dates so that your algorithms can easily detect whether it is the first or last bill segment for your SA. If the algorithm detects that this is the first or last bill segment for the service agreement, it can use the profile and/or TOU map start and end date/times on the service agreement rather than the calculated consumption period passed in.

Note. The interval pricing and TOU pricing algorithms provided with the product use the consumption period calculated and passed in from billing for the first and last bill segment.

Defaulting Service Agreement Time Options

In order to facilitate setup of your service agreements with their appropriate time options, cutoff time and start day option may be defined on an [SA Type Start Option](#).

Cutoff time and start day option must also be defined on the [installation option](#) record. If your service agreement is not created with a start option, the values on the installation record are used.

Changing Service Agreement Time Options

As described above, billing uses the service agreement's cutoff time and start day option to calculate the consumption period to the interval pricing algorithms. The system currently allows you to change the cutoff time and start day option in the middle of a contract, but this is very unusual. In fact, for a given company, these settings are often the same for all interval billing customers (or at least the same for an entire class of customers). For example, a company may designate that all transportation gas customers bill from 11pm to 11pm and all other interval billing customers bill from 2 a.m. to 2 a.m..

If this information changes in the middle of a contract and non-canceled bills exist, a warning is issued. Changing either of these fields after a bill segment has been generated will cause either a gap or an overlap in intervals billed for the next bill segment. A user must manually adjust the charges accordingly.

Time Zone and Time Changes

This section describes interval data considerations when operating in different time zones and when handling seasonal time changes, such as daylight savings time.

Contents

[Data is Stored in the Base Time Zone](#)
[Seasonal Time Shifts](#)

Data is Stored in the Base Time Zone

With data stored at an interval level, provisions need to be made to cater for data captured from other time zones.

- On the installation record, you indicate the base time zone for your company data.
- ALL data must be stored in Standard time for this base time zone.

In order to help interfaces to adjust data received from a different time zone, you may store the customer's time zone on its Premise. The interface may be written to check the premise's time zone, compare it to the base time zone, and shift the data accordingly.

Refer to [Designing Your Time Options](#) for more information.

Seasonal Time Shifts

A more common problem related to interval data is the problem of adjusting time for seasonal time changes. Interval data cannot be stored in "legal" time (i.e., in the summer, storing data in daylight savings time or summer time) because the shift from summer time back to standard time causes a duplicate hour. For example, in the United States, the 2 a.m. hour is repeated when shifting from Daylight Savings Time (DST) back to standard time.

Date/Time	Value	UOM	Local time as displayed
28/Oct/2000 21:00:00	2000	KW	28/Oct/2000 22:00:00
28/Oct/2000 22:00:00	2302	KW	28/Oct/2000 23:00:00
28/Oct/2000 23:00:00	234	KW	29/Oct/2000 00:00:00
29/Oct/2000 00:00:00	2352	KW	29/Oct/2000 01:00:00
29/Oct/2000 01:00:00	2525	KW	29/Oct/2000 02:00:00
29/Oct/2000 02:00:00	5324	KW	29/Oct/2000 02:00:00 DST -> Std, +1hr
29/Oct/2000 03:00:00	464	KW	29/Oct/2000 03:00:00
29/Oct/2000 04:00:00	25252	KW	29/Oct/2000 04:00:00
29/Oct/2000 05:00:00	252	KW	29/Oct/2000 05:00:00
29/Oct/2000 06:00:00	3453	KW	29/Oct/2000 06:00:00

To avoid the problem of duplicate records, all data must be stored in standard time. However, when entering and viewing data online, users probably want to see the data in the current "legal" time (i.e., in the summer, data is shown in daylight savings time or summer time).

Contents

[Logical Time versus Server Time](#)

Bill Period and Seasonal Time Shifts Interval Time Display Evenly Sized Intervals

Logical Time versus Server Time

As described in [Logical Time versus Server Time](#), fields may be defined as being system date / time stamps or logical data to be captured in standard time.

In the case of interval data:

- Each interval record has a Set Date / Time, which is considered server or physical time and uses the time shift information defined on the base time zone.
- The interval date/time is considered logical time and the time shift information is defined on the related “type” entity for the interval data.

Bill Period and Seasonal Time Shifts

As described in [Start and End Times for Billing](#), the service agreement cutoff time is used to determine the start and end times for billing.

This time is assumed to be in legal time, according to the seasonal time shift linked to the time zone for the service agreement’s characteristic premise. It means that when this service agreement is billed, the system adjusts the cutoff time to standard time prior to accessing the appropriate intervals. The diagram below illustrates this point.

Cutoff Time is 2 a.m.; Standard Time begins 22,Oct

Date/Time	Data
...	
01/Oct/2000 01:00:00	\$0.51
01/Oct/2000 02:00:00	\$0.52
01/Oct/2000 03:00:00	\$0.21
01/Oct/2000 04:00:00	\$0.21
...	
30/Oct/2000 00:00:00	\$0.51
30/Oct/2000 01:00:00	\$0.52
30/Oct/2000 02:00:00	\$0.21
30/Oct/2000 03:00:00	\$0.21
30/Oct/2000 04:00:00	\$0.21
...	

1st Oct. is during DST. The first interval is the one AFTER 2am legal time (1am standard time)

30th Oct. is during standard time. The last interval used is the one that includes the cutoff time (2 a.m.)

Interval Time Display

By default, the system displays interval data in legal time.

- For server related time fields, the time is displayed according to the seasonal time shift record on the base time zone.
- For interval related data, the time is displayed according to the seasonal time shift record on the interval entity’s type.
- For the effective date/time links between a service agreement and its interval collections, the time is displayed according to the seasonal time shift record on the SA’s characteristic premise’s time zone.

A message indicates this to the user, for example, "Date/Time Info is expected in local legal time". The user can opt to change the display to show the data in standard time.

When entering new data, the user is expected to enter data in the same time that is displayed. If the message states that intervals are expected in legal time, the system assumes that the user enters new values in the legal time and converts them to standard time for storing.

Note. If seasonal time shift information is missing for an entity or for the base time zone, data is always displayed and expected in standard time.

Evenly Sized Intervals

As mentioned earlier, the main drive behind shifting data to standard is to counter the effect of missing and duplicate intervals at entry to and exit from a seasonal time shift period. Storing data shifted to standard time ensures all interval records for a given curve are of equal size, which allows for simpler business logic, ignorant of seasonal time shifts considerations.

While this is obvious for hourly or less than an hour interval sizes, we would like to stress that the same concept applies to all interval sizes. If the legal time of an interval shifts during a seasonal time shift you should "counter-shift" it to standard time. If its legal time does not shift year-round, no shifting is needed.

For example, let's assume interval data being recorded in 1440 minute intervals (each read spans a full day).

- If when entering a seasonal time shift period, the legal time for these intervals shifts, say from midnight to 1 am, then when interfaced to the system data should be shifted back to standard to achieve evenly sized intervals.
- However, if the legal time does not shift, i.e. reads are taken at the same legal time, shifting is not applicable. As a matter of fact, shifting in this case causes the stored intervals to be of un-even size, thus complicating the logic that processes the data.

Base Plug-ins. If you decide to configure the system to not follow the evenly sized interval concept described above you may not be able to use the base sample plug-ins and common routines as they assume interval data is stored in evenly sized intervals.

Note. If you are using the interval entities but your interval data have interval sizes larger than hourly (for example, daily), you may decide not to use any seasonal time shift logic. You may set a switch on the installation record to indicate whether or not your interval data should observe seasonal time shifting. Refer to [Installation Options - Billing](#) for more information.

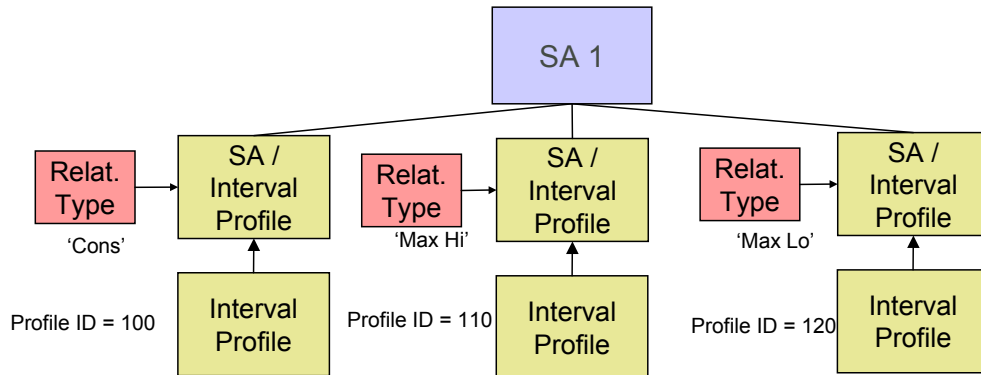
Interval Data Serves a Role for a Service Agreement

All interval quantities are linked to an Interval Profile. An interval profile may be linked to a service agreement to serve a variety of purposes, for example:

- It is used by rates algorithms to produce bill calculation lines.
- It is used by data derivation algorithms to produce other interval data.
- It is available to the company for statistical analysis or reports.

When linking an interval profile to a service agreement, the role that it is serving must be indicated. The “role” is specified using a profile relationship type. Refer to [Physical Attributes of Interval Data vs. Its Role](#) for more information.

The following diagram illustrates one service agreement with three different profiles linked to it. One profile is serving the role of Consumption, one is serving the role of Maximum High Demand, and one is serving the role of Maximum Low Demand.



If the rate for this service agreement bills for consumption, there is a rate component that references the **CONS** profile relationship type. Refer to [Setting Up Interval Pricing Rate Component](#) for more information.

Note. Profiles linked to a service agreement may or may not be billable. How do we know which ones are billable? We know because for billable profiles, the rate component for the SA's rate will indicate its profile relationship type.

It is also likely that data derivation algorithms will be written use the profile relationship type as parameters for determining the correct data to process.

Common Profiles vs. SA Owned Profiles

Interval Profiles are segregated into two categories:

- Profiles that are **SA Owned** are linked to a single “owner” SA. For example data related to the SA's interval meters. This data may be available for calculations on other service agreements. For example, maybe you have separate service agreements for distribution and transportation and both service agreements use the same data for calculations.
- Profiles that are **Common** typically contain data that is not related to a specific SA and can be used by many SAs. Some examples are:
 - Contract Demand or Subscribed Demand
 - Hedge Cover
 - Maximum Demand
 - Deemed (estimated) Profile

SA owned profiles reference a service agreement that is considered the owner of that data. Only **SA owned** profiles may reference data derivation algorithms. These algorithms are run for the owner SA.

Refer to [Interval Profile Maintenance](#) to understand how profiles are created and maintained.

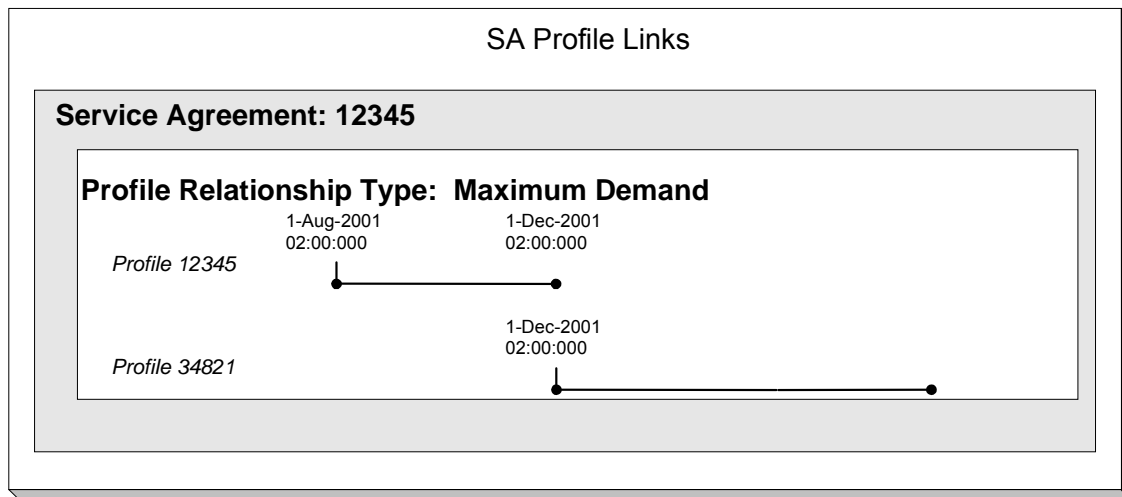
Flexibility of SA / Profile Link

The flexibility of the system allows for many combinations of profiles and service agreements:

- One profile may be linked to the same service agreement serving two different roles. Using the Profile Relationship Type to define the role, one profile could be, for example, the “Maximum Demand” and the “Minimum Demand” for a service agreement. This profile may be **common** or may be **SA owned**.
- One profile may be linked to two different service agreements. This profile may be **common** or may be **SA owned**. This profile may serve the same role for two different service agreements, or it may serve two different roles.

The Link between Profile and SA is Effective Dated

An SA/Profile link is effective dated. Therefore, the data serving a given role may change over time. To be more explicit, a service agreement may have the data for Profile A serving as the Maximum Demand interval values at the beginning of the contract, then after a few months, perhaps Profile B's data may be used instead. This scenario assumes that the contract itself did not need to change.



Why would there be a need to change the profile being used for a given role (especially when the data itself changes every few minutes anyway)? There could be several reasons:

- Using the Maximum Demand example, perhaps the customer's usage profile has changed and this change warrants a different Maximum Demand curve (although, the rate does not change). A different **common** profile needs to be linked to the service agreement.
- Perhaps the interval size of the data for the customer has changed. If the rate linked to the customer can cater for the new interval size, a new profile with a different profile type needs to be linked to the service agreement.
- Recall that a profile's type contains the creation and validation algorithms. Perhaps algorithms needed to create or validate this profile have changed. To cater for this a new profile with a different profile type must be linked to the service agreement.

This design also applies to the link between an SA and a TOU map.

Note. All algorithms that access profile and TOU map data must cater for the possibility of the profile or map changing during the desired period.

Creation of Profile Data through Data Derivation

As mentioned earlier, interval data linked to a profile may be interfaced to the system from an external source or it may be created by the system through a data derivation algorithm.

Contents

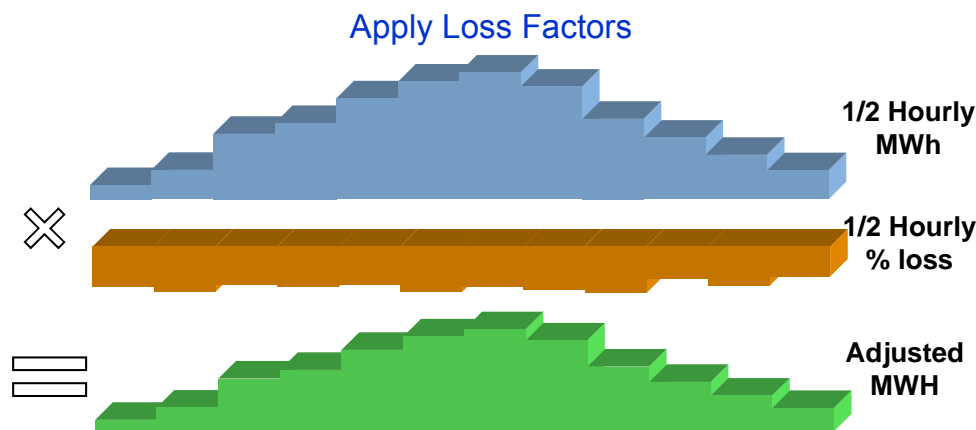
- [The Big Picture of Data Derivation](#)
- [Determining What Interval Data To Derive](#)
- [Force Derivation of Interval Data](#)
- [Process IB-SPDB - SA Interval Profile Data Creation](#)

The Big Picture of Data Derivation

Data derivation algorithms are used when a customer's contract requires interval data that is created based on other interval data in the system.

Some examples may include:

- Adjusting measured demand to account for loss factors to produce an adjusted billable data curve
- Comparing measured data to a contract demand curve to produce an adjusted billable data curve
- Comparing measured data to a maximum demand curve to produce an excess load curve
- Aggregating raw interval data from multiple interval registers linked to the service agreement's service points
- And more...



How is this set up in the system? Using the above loss factor example, let's make the following assumptions:

- The ½ Hourly MWh curve is an **SA Owned** profile linked to the service agreement. Its data is interfaced into the system from an external source. Refer to [Upload Interval Data](#) for more information.

- The ½ Hourly % loss are common for a group of customers and are stored as bill factor interval values. Refer to [Bill Factor Interval Values](#) for more information.
- The Adjusted MWh curve is an **SA Owned** profile linked to the service agreement. Its data is created by the system through a **Creation** data derivation algorithm linked to the [profile type](#).

The creation algorithm (that would need to be written to handle this example) would be provided with the profile relationship type of the ½ Hourly MWh curve and the Bill Factor for the loss factors. It applies the percentage loss factor for each interval and stores the result for each interval as a new data set for the Adjusted MWh curve.

These algorithms may be executed in one of the following ways:

- Through a background process that finds every SA that has an interval profile whose profile type indicates a creation algorithm. Refer to [Process IB-SPDB - SA Interval Profile Data Creation](#) for more information.
- There is a button available on the [SA Interval Info](#) page that allows a user to execute the data derivation algorithms for a single SA.
- There is a button available on the [Account Interval Info](#) page that allows a user to execute the data derivation algorithms for all the service agreements for an Account. The service agreements are processed in the [billing processing sequence](#) order defined on their [SA type](#).

Determining What Interval Data To Derive

Contents

[Preventing Gaps in Data](#)

[Using The Latest Updates to Source Data](#)

Preventing Gaps in Data

How do the derivation algorithms know what data needs to be derived? A simple answer is any data received since the last time data was derived should be included in the new derivation. This answer sounds good, but it's not as simple as that. Although we know the date and time of the last derivation, we don't know the date range of the data that was derived. A user may ask for data to be derived only through a certain date, even though data exists after that date. The next time data derivation runs, it may need to include the data that was after the cutoff for the previous derivation.

To illustrate this point, let's look at an example. Assume that a derivation algorithm takes a source curve and applies a value from a bill factor to arrive at the new curve. In the illustration below, data for the source curve exists up to 29th of October 2000 at 4 a.m. When deriving new data, the user enters a cutoff date of the 29th of October. The time used is the Cutoff Time on the service agreement, which in this case is 2 a.m. Refer to [Start and End Times for Billing](#) for more information.

The user runs the data derivation on the 31st of October at 10:02. That date and time are the Set Date/Time for the newly created data set.

Source Data through 29th Oct at 4am

Reads received 30/Oct/2000 12:39

Date/Time	Value	UOM
...		
28/Oct/2000 23:00:00	42	KW
29/Oct/2000 00:00:00	40	KW
29/Oct/2000 01:00:00	6	KW
29/Oct/2000 02:00:00	9	KW
29/Oct/2000 03:00:00	8	KW
29/Oct/2000 04:00:00	9	KW

Data derivation run on 31st Oct with a cutoff date/time of 29 Oct / 2am

Derivation 31/October/2000 10:02
Apply factor of 10 to each interval

Date/Time	Value	UOM
...		
28/Oct/2000 23:00:00	420	KW
29/Oct/2000 00:00:00	400	KW
29/Oct/2000 01:00:00	60	KW
29/Oct/2000 02:00:00	90	KW

The last 2 intervals were not included in this derivation

The next time data derivation runs, it is not possible for the algorithm to simply derive data that has been received since the last time data derivation was run. In the above example, only data received after the 31st of October will be derived. However, the last two intervals for the source data received on the 30th of October also need to be included.

The following diagram illustrates how the next derivation should behave.

Next data derivation run should include the intervals from the 'old' data set that weren't included in the last derivation

Reads received 30/Oct/2000 12:39

Date/Time	Value	UOM
...		
28/Oct/2000 23:00:00	42	KW
29/Oct/2000 00:00:00	40	KW
29/Oct/2000 01:00:00	6	KW
29/Oct/2000 02:00:00	9	KW
29/Oct/2000 03:00:00	8	KW
29/Oct/2000 04:00:00	9	KW

Derivation 1/December/2000 1:19
Apply factor of 10 to each interval

Date/Time	Value	UOM
29/Oct/2000 03:00:00	80	KW
29/Oct/2000 04:00:00	90	KW
29/Oct/2000 05:00:00	80	KW
29/Oct/2000 06:00:00	70	KW
...		
29/Nov/2000 01:00:00	60	KW
29/Nov/2000 02:00:00	70	KW

Reads received 30/Nov/2000 2:21

Date/Time	Value	UOM
28/Oct/2000 05:00:00	8	KW
28/Oct/2000 06:00:00	7	KW
28/Oct/2000 07:00:00	10	KW
...		
29/Nov/2000 01:00:00	6	KW
29/Nov/2000 02:00:00	7	KW
29/Nov/2000 03:00:00	7	KW

New data received later

Because users have control over what data may be derived (using the cutoff date), the algorithms cannot rely on data received since the last derivation.

To be safe and ensure that no gaps exist in the derived data, the algorithms should look for each interval from either the start of the SA or the start of the SA/Profile relationship through to the cutoff date/time to determine if new data should be derived. However, for performance reasons (and practical reasons) the algorithms supplied with the base product only look for new data to derive starting 45 days prior to the cutoff date. This is a soft parameter to the algorithms called **Number of Days to Process** and may be changed. Implementation specific algorithms may choose not to follow this example, but you must be sure to consider the performance implications.

Note. The derivation algorithms are provided both the SA start date and cutoff time and the SA/Profile date/time. The base algorithms derive data from the SA start date/cutoff time. Implementation specific algorithms have the option of deriving data from SA/profile date/time.

Refer to [interval profile type](#) for more information about the sample data derivation algorithms provided with the base product.

Using The Latest Updates to Source Data

As described in [Algorithms Find the Most Up To Date Data](#), the data derivation algorithms must also detect whether new data has been received for a given source interval that has already been used in derivation. For these cases, the more up-to-date data for the source interval should be used to re-derive the target curve.

Force Derivation of Interval Data

Derivation algorithms base their calculations on data from other profiles, TOU maps, registers, bill factors, contract options and so on. These may be referred to as the algorithms' source data.

Ideally, derivation algorithms should be able to detect changes made to their source data since the last derivation and re-derive values for the relevant periods if any changes are detected.

Refer to [Using the Latest Updates to Source Data](#) for more information.

As long as the source data has means of recording changes (e.g. an update date/time stamped on the new records), the algorithm can detect the changes and trigger re-derivation. However, not all objects in the system have this recording mechanism, making automatic detection of changes for re-derivation impossible.

Consider the sample derivation algorithm in [Preventing Gaps in Data](#). This algorithm used a bill factor to apply a constant to the source curve to arrive at a derived curve. Imagine that the users discover that the bill factor value was incorrect and that once this value is fixed, the data needs to be re-derived. The algorithms have no way of knowing that this bill factor has changed. In such cases, the user may run derivation and specify a period for which to force derivation, so that re-derivation with new values may take place. Refer to the background processes [Process IB-SPDB - SA Interval Profile Data Creation](#) and [Process IB-STDB - SA TOU Data Creation](#) for information about running derivation through batch. Refer to [Service Agreement - Interval Info](#) for information about running derivation online.

Note. The base derivation algorithms only support force derivation when there is source data other than interval data, such as a bill factor.

Process IB-SPDB - SA Interval Profile Data Creation

This process derives interval data for accounts in the system. Only accounts that have at least one interval service agreement with derivable profiles linked to it are processed. A 'derivable' profile is an **SA Owned** profile where this SA is the owner AND the profile type indicates an "Interval Data Creation" derivation algorithm. Interval data for service agreements linked to the account are derived in [billing processing sequence](#) order as defined on their [SA type](#).

Refer to [Determining What Interval Data To Derive](#) and [Start and End Times for Billing](#) for information about which records are processed by the algorithm.

For each service agreement, the interval data creation algorithms are executed in creation priority order. Refer to [Setting Up Interval Profile Types](#) for more information about defining Interval Data Creation algorithm for an interval profile type.

This process supports [Force Derivation](#) and passes the related input parameters to the data derivation algorithms.

This process is designed to run in parallel threads. Every thread processes a range of accounts.

Any errors detected during this process cause the new data set to be created in **Error** status and an entry to be written to the [Interval Data Exception](#) table. You can fix these errors by canceling the newly created data set and fixing the source of the problem. Refer to [How to Correct a Data Set in Error](#) for more information.

Validation of Profile Data

In addition to specifying algorithms to create data for an interval profile, you may also create algorithms to validate interval profile data. As with the creation algorithms, your validations algorithms are specified on the interval profile type. An interval profile type may specify more than one validation algorithm. If so, they are executed in priority order.

Any errors detected by the validation algorithms cause the invalid data set to be marked in **Error** status and an entry to be written to the [Interval Data Exception](#) table. You can fix these errors by fixing the source of the problem. Refer to [How to Correct a Data Set in Error](#) for more information.

These algorithms may be executed in one of two ways:

- Through a background process that finds every interval profile whose interval profile type specifies a validation algorithm. If multiple validation algorithms exist for a given profile type, they are executed one after the other in their predefined sequence order. Refer to [Process IPDSDBV – Interval Profile Data Validation](#) for more information.
- There is a button available on the [Interval Profile](#) page that allows a user to execute the validation algorithms for a single profile. If multiple validation algorithms exist for a given profile type, they are executed one after the other in their predefined sequence order.

Process IPDSDBV - Interval Profile Data Validation

The [IPDSDBV](#) background process is used to validate interval profile data. It processes pending interval profiles that were created up to the cutoff date/time and executes their validation algorithms, if any, defined on the profile type. The algorithms are executed in their predefined sequence order.

The standard business date is used together with the [installation base time](#) to derive the cutoff date/time that is passed to the validation algorithms. For example, if the batch business processing date is 09/08/2003 and the installation base time is 2:00AM, the cutoff date/time is calculated as 09/08/2003 02:00AM. When no business date is specified, it is set to current date.

It is important that ALL validation algorithms are provided with the same cutoff date/time as their referenced date/time. This identifies the collection of pending interval profiles to be processed as of a given date/time. This way even if a new profile is created after one validation algorithm is executed but before a subsequent validation algorithm is executed, the new profile is still not processed because it was not created by the cutoff date/time.

Refer to [Setting Up Interval Profile Types](#) for more information about defining Interval Data Validation algorithm for an interval profile type.

This process is designed to run in parallel threads. Each thread processes a range of profiles.

Refer to [Validation of Profile Data](#) for more information.

Auditing Your Interval Pricing Bill Lines

Given the amount of data used to produce a bill calculation line for an interval pricing rate component, how can a customer service representative audit this bill line to understand how it was calculated? Interval pricing rate components allow you to set up an Audit Algorithm. This audit algorithm retrieves the records used to produce the bill line. As a result, the audit algorithm should take advantage of the logic used in the calculation algorithm that produced the bill line. The relevant code should be shared between the two algorithms.

Refer to [Interval Billing Calculation Details](#) to view the results of calling an audit algorithm.

Refer to [Setting Up Interval Pricing Rate Components](#) for more information related to linking an audit algorithm to your rate component.

Warning! Snapshots of the data used to generate a bill are not taken. Rather, auditing your interval pricing bill lines rely on the interval data tables. As a result, **data records should not be changed**. To correct an interval, you should add a new interval data set to correct the interval data. Refer to [How To Correct a Data Set in Error](#) for more information.

The Big Picture of Raw Data Collection and Aggregation

The topics in this section describe the ability of the system to collect raw interval data, provide the ability to run validation algorithms on this data, and aggregate data for use on interval profiles used for billing. This functionality is sometimes referred to as “meter read pretreatment”.

Some organizations use an external system to collect and aggregate the raw interval data. This information is interfaced directly to the SA interval profiles as billable data. From there, contract based adjustments may be applied.

Some organizations use Oracle Utilities Customer Care and Billing to collect and aggregate their raw interval data. This information is interfaced to interval registers linked to the service point's meter. Validation algorithms may be applied against this data to ensure that the data is valid. Interval profile creation algorithms for the SA may aggregate the raw data from the appropriate service points to prepare for billing.

Note. If your organization uses an external system for this functionality, skip this section.

Contents

- [Interval Channels and Index Channels](#)
- [Installation of Interval and Index Channels](#)
- [Validation of Register Data](#)
- [Process IREGDVB - Interval Register Data Validation](#)
- [Processing Raw Data](#)

Interval Channels and Index Channels

Channel is a term often used for devices that may store data for complex metering customers. A physical channel may contain interval data or index readings:

- Index channels are a collection of time-of-use readings.
 - The information is often considered more accurate than interval data.
 - Index Channels are typically set up as a meter, where each register in the meter configuration represents a time of use code. Refer to [The Structure Of A Meter](#) for more information.
- Interval channels contain collections of interval data. This interval data follows the same logic as interval profile data. Refer to [Interval Quantities are Stored in Data Sets](#), [Fixing Errors with Complete Data Sets](#) and [Algorithms Find the Most Up To Date Data](#) for more information. An interval channel is set up as a register that is marked as "interval". Interval register data may be linked to this interval register.

Your setup of meters, meter configurations, and registers is very flexible:

- The meter in the system doesn't have to represent a physical meter.
- The meter represents a single channel or a collection of channels.
- Any of the following meter configurations are possible:
 - A single interval channel
 - Multiple interval channels
 - A collection of TOUs for an index channel
 - A combination of interval registers and TOUs for an index channel

Refer to [Maintaining Meter Configurations](#) for more topics related to meter configurations.

Note. In Oracle Utilities Customer Care and Billing, the term “channel” is only used for the Channel ID on the register, which is used as an external ID. In the system, you will use meters, meter configurations and registers to model your channels.

No special logic is required for index channels. The reads for the time of use collection for an index channel require the same logic that exists for standard non-interval registers.

Installation of Interval and Index Channels

Installing a configuration for an index or interval channel to a service point is the same procedure as installing a configuration of standard registers.

- You must define the meter configuration that is being installed.
- The installation is effective-dated. There must be a meter read on the install date.

Note that for interval registers, register reads are not allowed. However, you may create meter reads without register reads in order to install a meter configuration with interval registers on a service point.

Refer to [SP/Meter Installation](#) for more information about installing meters at service points.

Validation of Register Data

You may create algorithms to validate interval register data. Your validation algorithms are specified on the interval register type. An interval register type may specify multiple validation algorithms that are executed in priority order.

Any errors detected by the validation algorithms cause an invalid data set to be marked in **Error** status and an entry to be written to the [Interval Register Data Exception](#) table. You can fix these errors by fixing the source of the problem. Refer to [How to Correct a Register Data Set in Error](#) for more information.

These algorithms may be executed in one of two ways:

- Through a background process that finds every interval register with an interval register type that specifies a validation algorithm. If multiple validation algorithms exist for a given interval register type, they are executed in their predefined sequence order. Refer to [Process IREGDVB – Interval Register Data Validation](#) for more information.
- There is a button available on the Meter Configuration page that allows a user to execute the validation algorithms for all the interval registers linked to the meter configuration. If multiple validation algorithms exist for a given interval register type, they are executed one after the other in their predefined sequence order. Refer to [Meter Configuration](#) for more information.

Process IREGDVB - Interval Register Data Validation

The [IREGDVB](#) background process is used to validate interval register data. It processes pending interval registers that were created up to the cutoff date/time and executes their validation algorithms, if any, defined on their register type. The algorithms are executed one after the other in their predefined sequence order.

The standard business date is used together with the [installation base time](#) to derive the cutoff date/time that is passed to the validation algorithms. For example, if the batch business processing date is 09/08/2003 and the installation base time is 2:00AM, the cutoff date/time is calculated as 09/08/2003 02:00AM. When no business date is specified, it is set to current date.

It is important that ALL validation algorithms are provided with the same cutoff date/time as their referenced date/time. This identifies the collection of pending data sets to be processed as of a given date/time. This way even if a new data set is created after one validation algorithm is executed but before a subsequent validation algorithm is executed, the new data set is still not processed because it was not created by the cutoff date/time.

Refer to [Setting Up Interval Register Types](#) for more information about defining Interval Register Data Validation algorithm for an interval register type.

This process is designed to run in parallel threads. Every thread processes a range of registers.

Refer to [Validation of Register Data](#) for more information.

Processing Raw Data

The raw interval data linked to your registers is not used directly by rate algorithms. The rate algorithms only process profile data linked to the service agreement. Valid interval register data is available for use by the profile data derivation algorithms to produce billable profile data for a service agreement. For example, a profile data derivation algorithm may aggregate the register data from multiple registers to produce a new load curve.

A challenge for the aggregation algorithms is to determine which register data is applicable for a given profile. The service agreement may be linked to multiple service points, each with many different possible configurations. Because the derivation algorithm is on the interval profile type, you cannot explicitly tell the algorithm which registers to use. The following information may be useful to the algorithms to determine the appropriate registers:

- Service points linked to the service agreement. First and foremost, the algorithms restrict the selection of registers to those for the service points linked to the service agreement.
- UOM and SQI. Each interval register type indicates a unit of measure and an optional SQI that may be used by algorithms. For example, the base algorithm, which performs aggregation, finds interval registers with a UOM that matches that of the interval profile type. This algorithm also has soft parameters to indicate whether the SQI codes on the interval register type and interval profile type should match.
- If these attributes are not enough to indicate the correct registers to the algorithms, characteristics are also available for use. There are characteristics on the service point, the SA/SP and the meter that may all be used to capture information available to the processing algorithms to determine the appropriate registers.

Refer to [Designing Interval Profile Types](#) for more information about the aggregation algorithm provided with the system.

Applying Adjustments To Raw Data Prior to Aggregation

It is common for a contract to require adjustments to raw data prior to aggregating the data onto a profile. For example, perhaps all the data for a given register should be adjusted by 5% based on contract requirements. These adjustments may be applicable to only certain interval registers for the contract. In addition, it is common for more than one contract to use the same register data and perhaps only certain contracts require an adjustment to that data.

To satisfy this requirement, characteristics on the SA/SP may be used to indicate adjustment factors. The appropriate derivation algorithms may then apply the adjustment factors while aggregating the data.

Note. The aggregation algorithm provided with the system does not apply any adjustment factors.

Refer to [Service Agreement - SA/SP](#) for more information about SA/SP characteristics.

Time of Use Mapping Background Topics

This section describes information related to how data is stored for TOU Maps and how TOU Maps are linked to service agreements. For information about defining time of use codes and other related topics, refer to [Time of Use Billing](#).

For information about defining time of use rate components, refer to [Designing Your Time Of Use Rate Components](#). For information about defining TOU prices and values, refer to [Bill Factor TOU Values](#).

Contents

- [Time Period Definitions are Stored in Data Sets](#)
- [Sharing TOU Map Data](#)
- [TOU Maps Linked to a Service Agreement](#)
- [Customer Specific TOU Values](#)
- [Generating Data for a TOU Map](#)
- [Automatic Creation of TOU Map Data](#)
- [Auditing Your TOU Pricing Bill Lines](#)

Time Period Definitions are Stored in Data Sets

The time period definitions for a TOU map may have many different origins.

- It may be contract-related data that is set up at the beginning of the contract or contract year
- It may be interfaced from an external source periodically. This can occur when a customer's TOU map is defined dynamically based on information received from a third party.
- It may be that data needs to be overwritten for a given day. Perhaps the company has a planned interruption that changes the time period definitions for a given day. New data for the TOU map for that day needs to be linked to the map.

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

- Data is received or created by the system periodically. The system stores 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 more information about how interval data is stored. This same model is used for TOU data. There is no difference.

Refer to [Fixing Errors with Complete Data](#) for information about how to distinguish fixing errors with complete data by creating new data sets. This same model is used for TOU data.

Note. The system is able to generate data for a TOU map based on predefined templates. Refer to [Generating TOU data for a TOU Map](#) for more information.

Sharing TOU Map Data

TOU mapping data may be shared by more than one service agreement. There are several ways to accomplish this and your business rules will dictate the best way to do this.

Contents

- [TOU Map Common To All SAs on the Rate](#)
- [TOU Map Common To a Subset of Service Agreements](#)
- [TOU Map Owned by a Service Agreement](#)

TOU Map Common To All SAs on the Rate

If a TOU pricing rate component on the service agreement's rate performs TOU mapping and/or pricing, it is possible that all service agreements for the rate use the same **common** TOU map. In this case, you indicate the TOU map to use directly on the rate component. In this scenario, no map information is needed for the service agreement. Refer to [Designing Your Time Of Use Rate Components](#) for more information.

TOU Map Common To a Subset of Service Agreements

If not all service agreements linked to your rate use the same TOU map for a TOU pricing rate component, it may still be the case that a subset of those service agreements use the same TOU map. In this case, the appropriate TOU map for each service agreement must be linked to the appropriate **common** TOU map.

To set up these types of service agreements correctly, the appropriate start option should indicate the common TOU map to link to the SA. Refer to [Designing Your IB Start Options](#) for more information.

TOU Map Owned by a Service Agreement

Some of your service agreement may require unique map data. These types of TOU maps are typically used for customers whose map data is determined by factors related to the specific contract. It is possible for **SA owned** TOU map types to reference [data creation](#) algorithms. This allows the TOU map data to be overridden or determined dynamically.

While **SA Owned** TOU maps are linked to a single “owner” SA, it's possible for this TOU map to be used by other service agreements (typically ones that are linked to the same account.)

To set up these types of service agreements correctly, the appropriate start option should indicate the TOU map type to use when creating a new **SA Owned** TOU map to link to the SA. Refer to [Designing Your IB Start Options](#) for more information.

TOU Maps Linked to a Service Agreement

As mentioned above, if a service agreement's rate performs TOU mapping and/or pricing and the TOU map to use varies for different service agreements, the appropriate map must be linked to the service agreement.

There may be other reasons to link a TOU map to a service agreement. For example, a data derivation algorithm may require a TOU map for the service agreement.

The topics in this section describe other logic related to linking a TOU map to a service agreement.

Contents

- [TOU Maps Serve a Role for a Service Agreement](#)
- [The Link between TOU Map and SA is Effective Dated](#)

TOU Maps Serve a Role for a Service Agreement

A TOU map may be linked to a service agreement to serve a variety of purposes, for example:

- Rate algorithms map interval quantities in order to apply prices for each TOU.
- Data derivation algorithms compare quantities to TOU values in order to validate the data or produce a new curve.

When linking a TOU map to a service agreement, the role that it is serving must be indicated. The “role” is specified using a TOU Map Relationship Type.

This is the same model as the one used to link interval profiles to service agreements. Refer to [Interval Data Serves a Role for a Service Agreement](#) for more information.

If a TOU pricing rate component indicates a TOU map relationship type, it is used to find the correct TOU map for the service agreement for mapping the interval data. Refer to [Setting Up TOU Mapping Rate Components](#) for more information.

The Link between TOU Map and SA is Effective Dated

An SA/TOU Map link is effective dated. Therefore, the TOU map serving a given role may change over time.

This is the same model as the one used to link interval profiles to service agreements. Refer to [The Link between Profile and SA is Effective Dated](#) for more information.

Customer Specific TOU Values

As described in [Bill Factor TOU Values](#), a bill factor may define prices or values for a collection of time of use codes. A TOU bill factor may also indicate that the value can be found in a contract term. This enables you to define customer specific time-of-use prices or contract values.

The TOU Contract Values page enables you to define a set of values for a collection of time-of-use codes. The collection is effective dated. This data is available for use by rate component algorithms that apply prices for time-of-use quantities or data derivation algorithms that need contract based TOU values to perform derivation. Refer to [Service Agreement - TOU Contract Values](#) for more information about defining TOU Contract Values.

Generating Data for a TOU Map

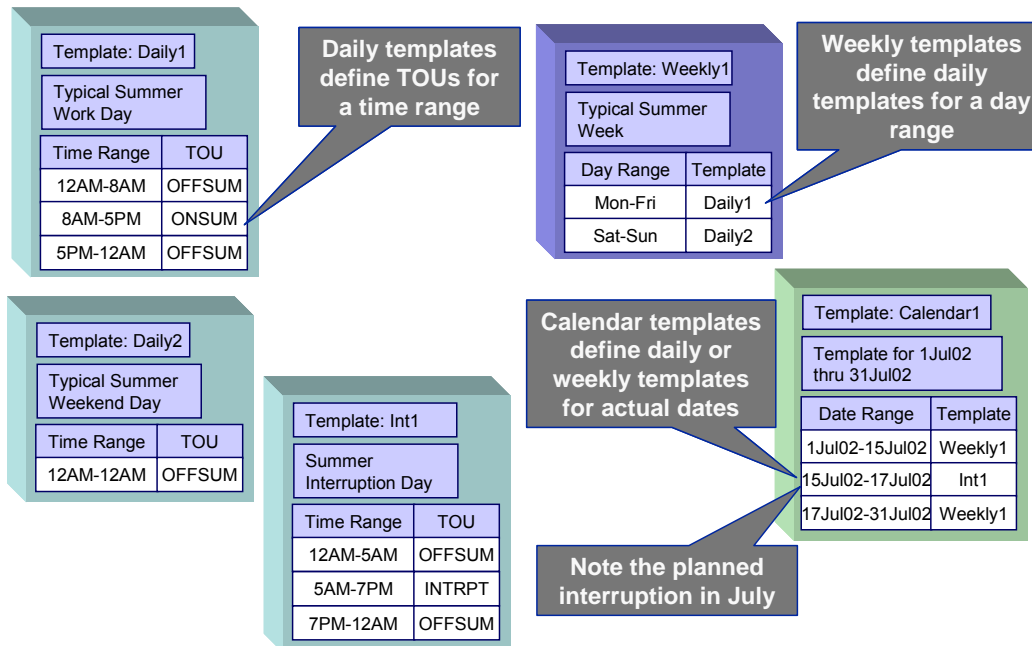
The data for a TOU map is a collection of date and time intervals and their corresponding time-of-use codes. Even though you may have several consecutive time periods referencing the same time-of-use code, each interval must be separately defined.

In order to facilitate the creation of data for a TOU map, the system enables you to define TOU map templates and provides you with the ability to generate data for a TOU map based on this template.

Actually, you may define a series of templates that work together to produce data for a TOU map. For example:

- Use a daily template to define time period ranges in a day with their associated TOU codes.
- Use a weekly template to define days of the week and the appropriate daily templates to use for each day.
- Use calendar template to define the weekly and daily templates applicable for actual dates in a year.

The following diagram illustrates a simple example of a collection of templates you may define for a summer period.



In the above example, a US resident may notice that no holiday is defined for the 4th of July. One would expect that if a holiday falls on a weekday, the typical weekday time period definitions may not apply.

The system is designed so that you don't have to define holidays in a TOU map template with their explicit dates.

- You have already defined which days of the year are holidays on the Calendar table.
- If all your holidays follow the same time period definition requirements, define a daily template that should be used for holidays.

When generating data for a TOU map, you are asked for:

- The appropriate calendar where your holidays are defined
- The holiday template

The TOU data generator automatically uses the holiday map for holidays defined in the calendar table.

Refer to [Setting up TOU Map Templates](#) for more information about defining your TOU map templates.

Refer to [TOU Map Generation](#) for more information about generating data for a TOU map based on a template.

Automatic Creation of TOU Map Data

The section [Generating Data for a TOU Map](#) describes how TOU map templates may be set up and used to generate TOU map data. When defining the appropriate TOU map for a given contract, you typically generate the TOU map data at the beginning of the contract year.

However, for many contracts, you may define special situations, such as interruptions, when an override TOU map is applicable for a certain time period. In fact, you may have a group of customers, who are all affected by the same interruption. The system allows you to configure your interval billing options such that once your interruption period has been defined an override TOU map for that interruption period can be generated for all customers.

In order to generate TOU map data, you need to design and create an appropriate algorithm. TOU map creation algorithms are linked to the TOU map type. You also need to define the "special situation", for example, the interruption.

Refer to [Contract Option Background Topics](#) for more information about special events that may alter the rate calculations.

In addition, refer to [Interval Billing Examples - Override Maps](#) for an example of how to set up an interruption example.

TOU map creation algorithms may be executed in one of the following ways:

- Through a background process that finds every SA that has a TOU map with a TOU map type that indicates a creation algorithm. Refer to [Process IB-STDB - SA TOU Data Creation](#) for more information.
- There is a button available on the [SA Interval Info](#) page that allows a user to execute the TOU data derivation algorithms for a single SA.

- There is a button available on the [Account Interval Info](#) page that allows a user to execute the TOU data derivation algorithms for all the service agreements for an account. The service agreements are processed in the [billing processing sequence](#) order defined on their [SA type](#).

Contents

[Determining What TOU Data To Derive](#)
[Force Derivation for TOU Map Data](#)
[Process IB-STDB - SA TOU Data Creation](#)

Determining What TOU Data To Derive

Refer to [Determining What Interval Data To Derive](#) for detail about ensuring that no gaps exist and ensuring that the algorithms are always using the latest data for derivation.

Note one difference between the base TOU creation algorithms and the base interval derivation algorithms is that the TOU creation algorithms do not use a **Number of Days to Process**. This is because the base algorithm processes contract option events and the volume of contract option events in the system is not as high as the volume processed by the interval data derivation algorithms.

Force Derivation for TOU Map Data

Refer to [Force Derivation of Interval Data](#) for information on how the system may allow a user to force derivation for situations where the system may not be able to detect that source data has changed. The same logic is available for TOU map creation algorithms.

Note. The source data for the base TOU map creation algorithms is always contract option data and as a result, there is no need to support Force Derivation in those algorithms.

Process IB-STDB - SA TOU Data Creation

This process derives TOU map data for accounts in the system. Only accounts that have at least one interval service agreement with derivable TOU maps linked to it are processed. A 'derivable' map is an **SA Owned** map where this SA is the owner AND the map type indicates a "TOU Data Creation" algorithm. TOU map data for service agreements linked to the Account are derived in [billing processing sequence](#) order as defined on their [SA type](#).

For each service agreement, the TOU Map Creation algorithms are executed in creation priority order. Refer to [Setting Up TOU Map Types](#) for more information about defining a TOU Data Creation algorithm for a TOU map type.

This process supports [Force Derivation](#) and passes the related input parameters to the data creation algorithms.

This process is designed to run in parallel threads. Each thread processes a range of accounts.

Any errors detected during this process cause the new data set to be created in **Error** status and an entry to be written to the [TOU Data Exception](#) table. You can fix these errors by canceling the newly created data set and fixing the source of the problem. Refer to [How to Correct a TOU Data Set in Error](#) for more information.

Auditing Your TOU Pricing Bill Lines

Just as with interval pricing rate components, TOU pricing rate components may also use an audit algorithm to enable a user to view the details of the calculations.

Refer to [Interval Billing Calculation Details](#) to view the results of calling an audit algorithm.

Refer to [Setting Up TOU Pricing Rate Components](#) for more information related to linking an audit algorithm to your rate component.

Warning! Snapshots of the data used to generate a bill are not taken. Rather, auditing your TOU pricing bill lines rely on the interval data and TOU mapping data tables. As a result, **data records should not be changed**. To correct data, you should add a new data set. Refer to [How To Correct a Data Set in Error](#) for more information.

Contract Option Background Topics

This section describes information related to contract options under which certain calculations supporting a contract's rate may be overridden or altered occasionally for specific periods of time. Also refer to [Designing Your Contract Options](#) for more information.

Contents

- [Overview](#)
- [A Contract Option Has Events](#)
- [Options and Events May Have Characteristics](#)
- [SA May Override a Shared Event](#)
- [Validation of Events](#)

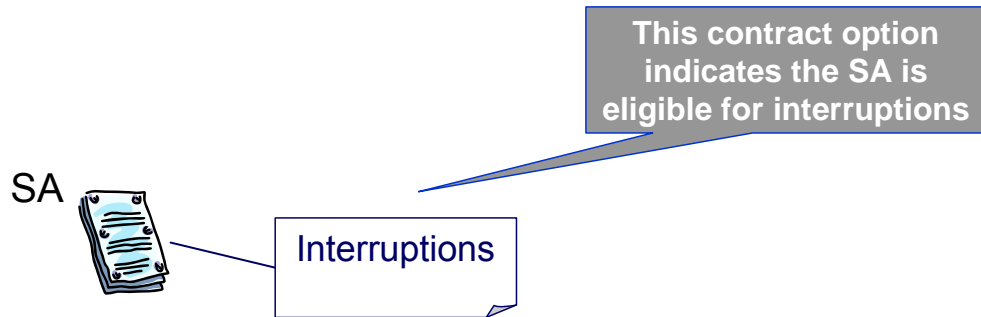
Overview

For some of your interval billing customers, you may define additional options under which certain calculations supporting the customer's rate may be overridden or altered occasionally for specific periods of time. Some examples of such options are:

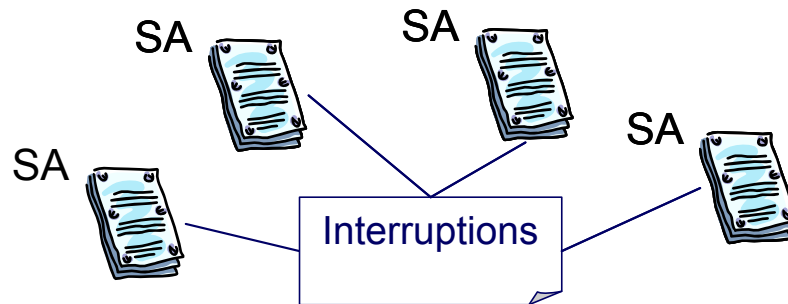
- Special days throughout the year on which the contract's TOU map is overridden with a special predefined TOU pattern specified for that type of day
- Periods of time during which customers are required to lower their demand by a predefined contractual amount of power
- Occasional offers to pay a special price for energy falling within a specific demand range

A Contract Option Has Events

A contract option must be defined for each special option on a given rate. The contract option is linked to every service agreement eligible for this option.

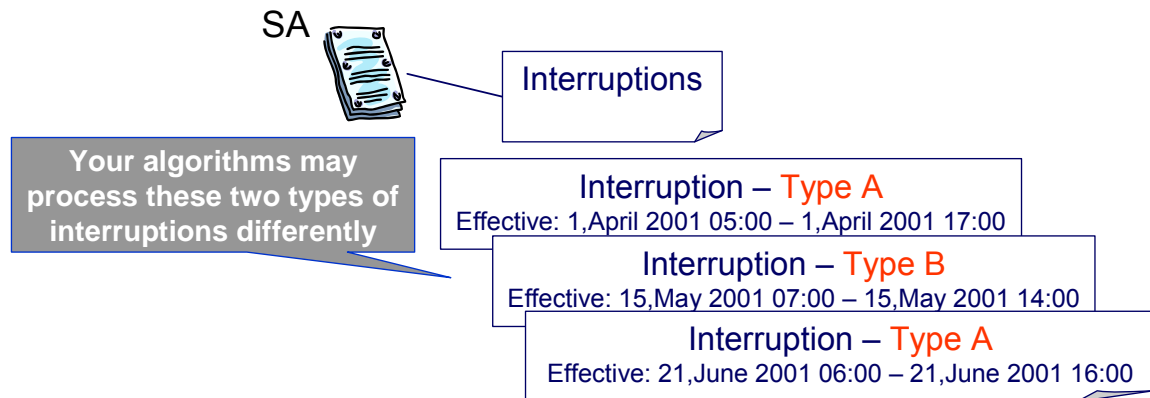


The option may be related to a given service agreement or may be linked to many service agreements.



Refer to [Contract Option Maintenance](#) for more information about creating contract options and refer to [Service Agreement - Contract Option](#) for more information about linking contract options to service agreements.

Each contract option contains a collection of contract option events. The events are the individual instances of the special option. Each event indicates its effective period. A given contract option may define different types of events. For example, perhaps you have several types of interruptions that should cause different overrides to occur. It is the responsibility of your data derivation algorithms and/or your rate algorithms to know how to process these events for a customer who may have contract options.



Refer to [Contract Option Event Maintenance](#) for more information about creating contract option events.

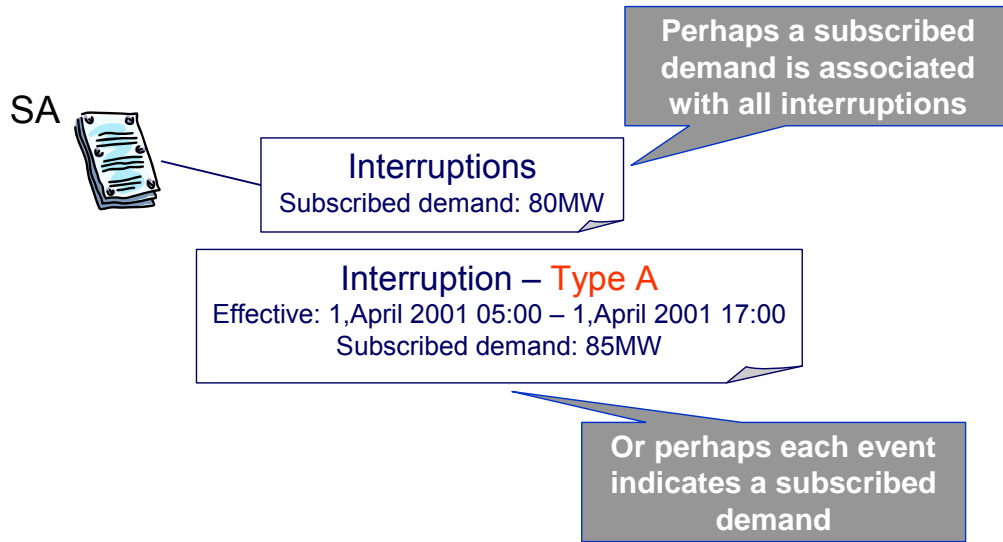
The following are some examples of algorithms that may process contract option events:

- You may have a TOU map creation algorithm that creates override TOU map data for the period defined on a contract option event. This override map may define different time-of-use codes for a certain time period, where different prices are applicable or may alter the time period definitions for this special time periods. The standard TOU pricing rate component would then apply the appropriate prices for this override time period.
- You may have a TOU pricing rate component that only applies a charge if contract option events exist for billing period. For example, it may calculate a special charge based on whether a customer reduced their demand during a specific period.

For more examples, refer to [Interval Billing Examples](#).

Options and Events May Have Characteristics

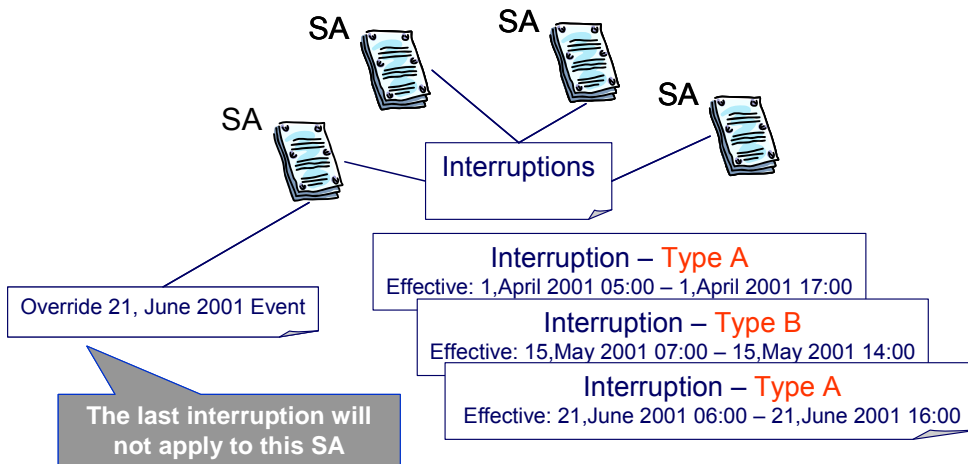
It is possible that your algorithms require extra information related to a contract option or a contract option event. For example, perhaps the customer's demand during a contract option event period must be compared to a subscribed demand. Perhaps the subscribed demand is applicable to all customers linked to this contract option. Or perhaps the subscribed demand changes for each instance of the option, in other words for each event.



Characteristics are used to define this extra information. Your algorithms would then use the information in the characteristic collection for either the contract option or the contract option event during its processing.

SA May Override a Shared Event

There may be cases where an individual customer wants to override the effects of a shared contract option event. If a given contract option event is not applicable to one of the service agreements linked to the contract option, a contract option event override may be created for this service agreement. The existence of an override cancels the effects of the shared contract option event for that SA.



You may also cancel an override, which essentially reinstates the specific event for the service agreement.

Note. Algorithms that process contract option events must process the service agreement overrides as well. This includes processing overrides that are subsequently canceled.

Refer to [Service Agreement - Contract Option](#) for more information about entering contract option overrides.

Validation of Events

The system provides you with the ability to run validation algorithms to validate your events. The validation algorithms may be used, for example, to validate the characteristics linked to your events. You may define separate validation algorithms that are executed based on the status:

- Pending algorithms are executed when adding or changing a record in **Pending** status.
- Freeze algorithms are executed when attempting to change the status from **Pending** to **Freeze**.
- Cancel algorithms are executed when attempting to change the status from **Freeze** to **Cancel**.

Refer to [Designing Your Contract Option Types](#) for more information about contract option event algorithms.

Maintaining Interval Data

In this section, we describe the pages that maintain interval billing and TOU mapping information.

Contents

- [Account Interval Info Maintenance](#)
- [Interval Profile Maintenance](#)
- [Maintaining Interval Profile Data](#)
- [Maintaining Interval Register Data](#)
- [TOU Map Maintenance](#)
- [Maintaining TOU Map Data](#)
- [Contract Option Maintenance](#)
- [Maintaining Contract Option Events](#)
- [Interval Billing Calculation Details](#)

Account Interval Info Maintenance

Open **Main Menu**, **Customer Information**, **Account Interval Information** to view interval profiles and TOU maps for service agreements linked to the account.

Note. This page is only available if the **Complex Billing** module is not [turned off](#).

Description of Page

The **Account Info** displays information about the account. The **Account ID** for this account is displayed.

Use the **Derive Interval Data** button to [create profile data through data derivation](#). If clicked, the **Derive Interval Data** dialog opens.

In the resulting dialog, turn on [Force Derivation](#) and enter a **Force Derivation Start Date** if you want the system to re-derive data for a period that may have already been derived.

Note. Not all algorithms necessarily support [force derivation](#). It depends on whether or not the algorithm can detect changes in the source data.

Enter the **Cutoff Date** to indicate date until which to generate data.

After clicking **Process**, the system processes all service agreements for the account that have an **SA Owned** profile where this SA is the owner AND the [profile type](#) indicates an “Interval Data Creation” derivation algorithm. Interval data for service agreements linked to the Account are derived in [billing processing sequence](#) order as defined on their [SA type](#). For each service agreement, the Interval Data Creation algorithms are executed in creation priority order as defined on their profile type.

The interval profile [tree](#) displays information about profiles linked to service agreements for this account.

- A node appears for each profile linked to more than one service agreement for the account. The profile may be **common** or **SA owned**. Expanding this node displays each service agreement linked to the shared profile.
- A node appears for each service agreement linked to the account that has interval profiles linked to it. Expanding this node displays each profile linked to the service agreement.

Use the **Derive TOU Data** button to [automatically create TOU map data](#). If clicked, the **Derive Interval Data** dialog opens.

In the resulting dialog, turn on [Force Derivation](#) and enter a **Force Derivation Start Date** if you want the system to re-derive data for a period that may have already been derived.

Note. Not all algorithms necessarily support [force derivation](#). It depends on whether or not the algorithm can detect changes in the source data.

Enter the **Cutoff Date** to indicate the date until which to generate data.

After clicking **Process**, the system processes all service agreements for the account that have an **SA Owned** TOU map where this SA is the owner AND the [TOU map type](#) indicates a “TOU Data Creation” derivation algorithm. Interval data for service agreements linked to the Account are derived in [billing processing sequence](#) order as defined on their [SA type](#). For each service agreement, the TOU Data Creation algorithms are executed in creation priority order as defined on their TOU map type.

The TOU map [tree](#) displays information about TOU maps linked to service agreements for this account.

- A node appears for each TOU map linked to more than one service agreement for the account. This TOU map may be **common** or **SA owned**. Expanding this node displays each service agreement linked to the shared TOU map.
- A node appears for each service agreement linked to the account that has TOU maps linked to it. Expanding this node displays each TOU map linked to the service agreement.

If a seasonal time shift record is linked to the base time zone and each time zone on the characteristic premise for each service agreement uses the same seasonal time shift record, the common **Seasonal Time Shift** for the time zones on the characteristic premises is displayed. Refer to [Time Zone and Time Changes](#) for more information.

The **Seasonal Time Shift Remark** indicates whether the data is displayed in legal time or standard time. If the seasonal time shift record on the base time zone and for the time zone for each characteristic premise linked to each SA are the same, the dates are displayed in legal time, according to this seasonal time shift record. Otherwise, the data is displayed in standard time.

Refer to [Seasonal Time Shifts](#) for more information.

Interval Profile Maintenance

Use this page to view information about a profile, its data sets and its service agreements.

Contents

[Interval Profile - Main](#)

[Interval Profile - Related SAs](#)

Interval Profile - Main

Open **Main Menu**, **Interval Billing**, **Interval Profile** to maintain interval profiles.

Note. This page is only available if the **Complex Billing** module is not [turned off](#).

Description of Page

The **Interval Profile Info** contains important information about the interval profile. These values only appear after the interval profile exists on the database. The **Interval Profile ID** is a system-assigned number.

Enter the **Description** of the Interval Profile.

Indicate the **Interval Profile Type** that defines this interval profile.

The **External ID** is available for cross-referencing this interval profile with an external source.

If this profile is **SA Owned**, the **SA ID** linked to this profile is displayed along with information about the SA.

Use the **Complete Data Sets** button if you have any pending data sets for your profile and you would like the system to complete them. When clicking this button, if the profile type contains any validation algorithms, they are executed. If no errors are detected from the profile type's validation algorithms OR if the profile type does not contain any algorithms, the **pending** data sets are marked as **complete**. Refer to [Validation of Profile Data](#) for more information about the validation algorithms.

The [tree](#) at the bottom of this page shows information about data sets linked to the profile.

- The **Pending Data Sets** node displays if any of the data sets linked to the profile are **pending**. Expanding this node displays each **pending** data set.

- The **Error Data Sets** node displays if any of the data sets linked to the profile are in **error**. Expanding this node displays each **error** data set.
- The **Data Created Through** node displays the date of the latest complete interval as of the current date. Expanding this node displays the last 10 **complete** data sets.

Interval Profile - Related SAs

Open **Main Menu**, **Interval Billing**, **Interval Profile** and navigate to the **Related SAs** tab to view service agreements to which this profile is linked.

Note. This page is only available if the **Complex Billing** module is not [turned off](#).

Description of Page

The **Interval Profile Info** contains important information about the interval profile. These values only appear after the interval profile exists on the database. The **Interval Profile ID** is a system-assigned number.

Note. If the profile is linked to more than 25 service agreements, the search criteria are intentionally left blank in order to avoid retrieving all service agreements (with the resultant slow response times). You must therefore use the **SA Filter** to define the type of service agreements that should be retrieved. See the below for more information about this page's search criteria.

Use the **SA Filter** to define the service agreements to 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. You can specify any combination of these fields.
- **All.** Use this option if you do not want to filter service agreements.
- **Person Name.** Use this option to restrict service agreements to those whose account indicates the given **Name** as the main person.
- **SA Type.** Use this option to restrict service agreements to those linked to a given **CIS Division** and **SA Type**.

Don't forget to click the search button after changing the **SA Filter**.

The grid that follows contains the **Accounts** and **Service Agreements** that match your search criteria.

Maintaining Interval Profile Data

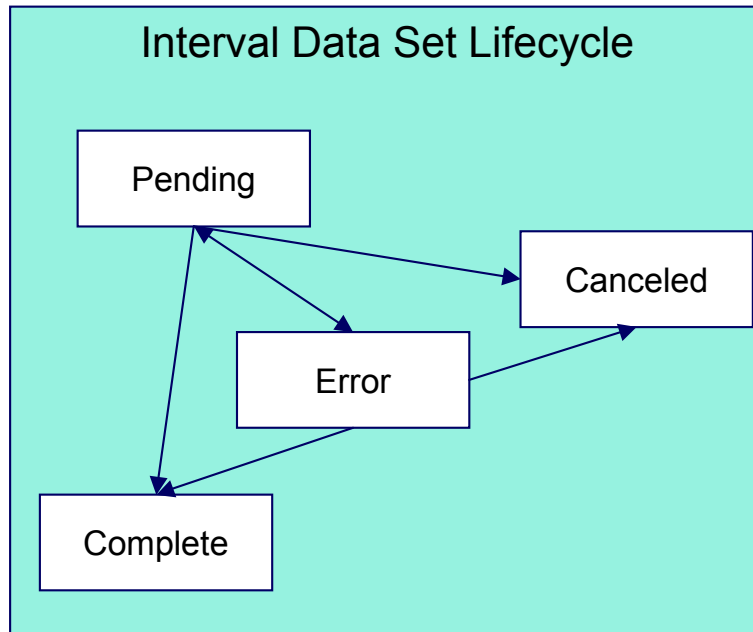
In this section, we describe information related to maintaining interval data. This information includes where to maintain the data, the life cycle that this data may go through, uploading data from an external source and handling exception conditions.

Contents

[Life Cycle of Interval Profile Data Sets](#)
[Interval Data Maintenance](#)
[Interval Data Query](#)
[Upload Interval Data](#)
[Process IPDSIDB - Determine Profile For Profile Datasets](#)
[Interval Data Exception](#)

Life Cycle of Interval Profile Data Sets

The following diagram shows the possible lifecycle of an interval profile data set.

***Pending***

Use this status for data, added to the system, that requires validation to occur before it is available for further processing. Validation is handled through an algorithm linked to the profile type for this data set's profile. Refer to [Setting Up Interval Profile Types](#) for more information. When creating a data set, the system automatically sets the status to ***Pending*** when the profile type indicates a validation algorithm or if an interval profile has not been identified.

Canceled

A user can cancel a ***Pending*** or ***Error*** data set if the data set should not be used for further processing. Refer to [Interval Data Maintenance](#) for more information. Additionally, algorithms may be designed to cancel data sets in certain situations.

Error

A validation or derivation algorithm may set this status for a data set if it detects an error condition. Data sets in this status are also written to the [Interval Data Exception](#) table. Refer to [How To Correct a Data Set in Error](#). When a validation error has been corrected, change the status of the data set back to **Pending** so it will be reprocessed.

Complete

Only complete data sets are used by rate application and derivation algorithms. If data being added to the system has already been validated, it may be uploaded to the system in this status. Data derivation algorithms may create a data set in **Complete** status or change the status of a data set to **Complete** if no error conditions are detected. Data validation algorithms may change the status of a data set to **Complete** if no error conditions are detected. When creating a data set, the system automatically sets the status to **Complete** when the profile type for the interval profile does not indicate a validation algorithm.

A data set is created either in **Pending** status or **Complete** status, depending on the existence of a validation algorithm on the profile type. A user can cancel a dataset. Other status changes may depend on how your validation and creation algorithms are written.

- The base validation algorithm only processes **Pending** data sets. As a result, a user is expected to change the status of an error data set back to pending when the error condition has been fixed. When the data is clean, the algorithm changes the status to **Complete**.
- The base creation algorithms automatically delete any **Error** data sets and start over in an attempt to produce a **Complete** data set. This is similar to the billing process, which deletes error bill segments and tries from the beginning to produce an error free segment. Another alternative is for a user to **Cancel** any data sets in **Error** to keep a record of the incorrect data.

For more information about data derivation and validation algorithms, refer to [Setting Up Interval Profile Types](#).

Interval Data Maintenance

The Interval Data page is used to create and maintain collections of interval data, grouped together by an interval data set, for a given interval profile. The data that is maintained here may be for a **common** interval profile or for an **SA owned** interval profile.

Open **Main Menu, Interval Billing, Interval Data** to maintain interval data.

Note. This page is only available if the **Complex Billing** module is not [turned off](#).

Description of Page

Interval Data Set is a concatenation of important information about the data set. The **Interval Data Set ID** is a system-assigned random number.

Indicate the **Interval Profile ID** to which this data set is linked.

Note. It is possible that the profile cannot be identified when adding interval data. If the **Interval Profile** is not populated, the **Data Set External ID** must be populated. Refer to [Upload Interval Data](#) for more information.

The **Owner SA** indicates the SA that “owns” this data. This information is only populated for **SA owned** profiles. This field is useful if the data set you are viewing contains an error and the error is related to a different source profile for the service agreement. You may use the context button to quickly navigate to the service agreement to view other profiles related to the service agreement. Refer to [how to correct a data set in error](#) for more information.

The **Data Set External ID** is the external identifier of the interval profile. It may be used when a profile cannot be found for data interfaced from outside the system. Refer to [Upload Interval Data](#) for more information.

The **Set Date/Time** of the Interval Data Set represent when this collection of data was entered or received. This is considered a server date and follows the seasonal time shifting as defined on the base time zone. Refer to [Seasonal Time Shifts](#) for more information.

Refer to [Algorithms Find the Most Up-To-Date Data](#) for more information about the Set Date/Time and its use in algorithms.

The **Data Set Type** is available for classification of the data set in some way. The value currently available with the system is **No Data**. Refer to [Fixing Errors with Complete Data](#) for more information about using this data set type.

Note. The values for Data Set Type are customizable using the [lookup table](#). This field name is INTV_DS_TYPE_FLG.

The **Status** indicates the status of the interval profile data set. Click the **Cancel DS** button to cancel a **Pending** or **Error** data set.

Refer to [Lifecycle of Interval Profile Data](#) for more information about interval data set status.

Use the **Revert** button to reverse the effect of a **Complete** data set.

Refer to [Fixing Errors with Complete Data](#) 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 Data Date/Time** and the **Interval Data** value are captured. This date is considered a logical date and follows the seasonal time shifting as defined on the interval profile type. Refer to [Seasonal Time Shifts](#) for more information.

If the data set is in **Error** status, a **Message** that provides the error information is displayed below the interval data collection.

If seasonal time shift records are linked to both the base time zone and the interval profile type, the **Seasonal Time Shift** associated with your interval profile type is displayed. Refer to [Setting Up Interval Profile Types](#) for more information.

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 does not indicate a seasonal time shift record, the toggle button is not visible and data is displayed and expected in standard time.

If seasonal time shift records are linked to both the base time zone and the interval profile type, 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 interval profile type or the base time zone does not indicate a seasonal time shift record, the toggle button is not visible and data is displayed and expected in standard time.

Interval Data Query

This query shows the most recent interval profile data for a profile and a time period as of a given time.

Open **Main Menu, Interval Billing, Interval Data Query** to display interval data.

Note. This page is only available if the **Complex Billing** module is not [turned off](#).

Description of Page

The **Interval Profile ID** is the identifier of the profile whose data should be 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 are accessed.

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

History

This option displays all interval data for the profile and time period, if specified. It includes data in data sets of any status value.

Most Recent Comp. Exc. No Data

This option finds the most recent interval value for each interval from completed data sets only. If the most recent interval is in a **No Data** data set, that interval does not appear.

Most Recent Complete Data

This finds the most recent interval value for each interval from completed data sets only. If the most recent interval is in a **No Data** data set, that interval appears with “No Data” displayed in the **Data Set Type** column.

Refer to [Fixing Errors with Complete Data](#) for more information about **No Data** interval data sets.

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

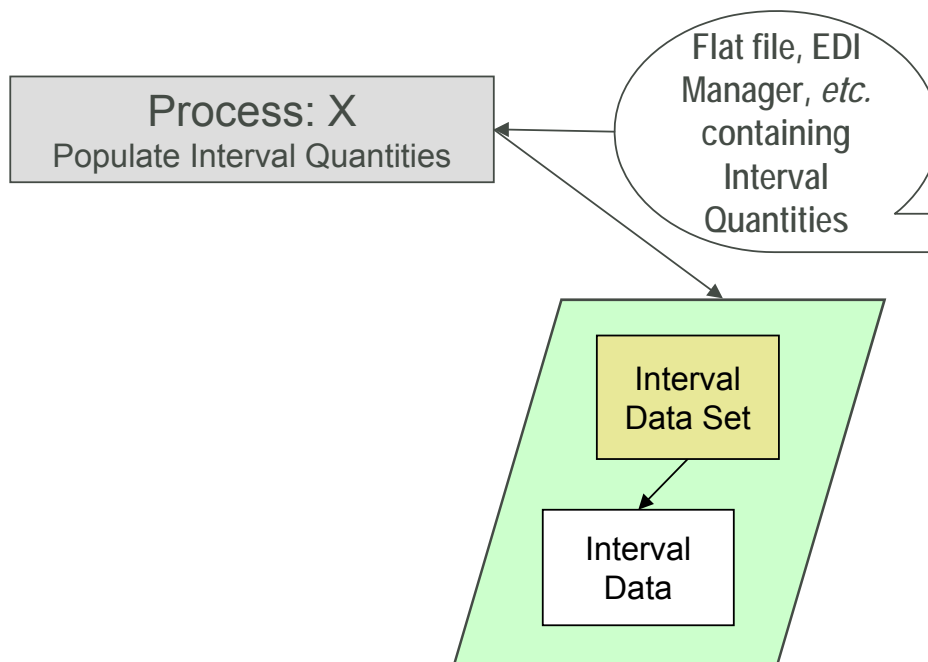
The **Interval Data Date/Time** and **Interval Data** value for each interval is displayed. Adjacent, information about the data set to which this interval is linked is displayed, including the **Data Set Type**, **Set Date/Time** and **Interval Data Set ID**.

If seasonal time shift records are linked to both the base time zone and the interval profile type, the **Seasonal Time Shift** associated with your interval profile type is displayed.

If seasonal time shift records are linked to both the base time zone and the interval profile type, 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 interval profile type or the base time zone do not indicate a seasonal time shift record, the toggle button is not visible and data is displayed and expected in standard time.

Upload Interval Data

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



The topics in this section describe how these processes work.

Contents

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

Process X - Populate Interval Data

Process X refers to the mechanism used by your organization to populate the Interval Data tables. Process X has the following responsibilities:

- It attempts to determine the appropriate profile to which this data belongs using the external ID. If a profile with this external ID is not found, the data set is created without being linked to a profile and the external ID is stored on the data set.
- It creates a new interval data set 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 standard time in the base time zone. Refer to [Time Zone and Time Changes](#) for more information.
- All quantities being uploaded for the same profile should be inserted into the Interval Data table and should reference the same interval data set.
- Set the status of the new data set:
 - If the data set is not linked to a profile, set the status to pending.
 - If the profile type has a validation algorithm, set the status to pending.
 - If the profile type does not have a validation algorithm, this data should be clean. Set the status to complete.
- Inserts a record in the interval data set key table.

The topics in this section describe the tables to populate.

Interval Data Set

You must create a data set for each batch of records being uploaded in the system for the same profile. The name of this table is [CI_INTV_DATA_SET](#). The following table describes each column on this table.

Column Name	Length	Req'd	Data Type	Comments
INTV_DATA_SET_ID	12	Y	A/N	<p>This is a unique key. This key is referenced on the interval data record.</p> <p>It's suggested that this key is built as follows:</p> <ul style="list-style-type: none"> • The first five digits are the first five digits of the INTV_PF_ID • The last seven digits are randomly generated. <p>This facilitates set based processing of the data. If your process that inserts records on this table is capable of calling a COBOL routine, call CIPZKEYG passing the first five digits and it supplies a seven digit random number.</p>
INTV_PF_ID	12	N	A/N	<p>This is a foreign key reference to the interval profile to which this set is linked. It is required if an external ID is not specified.</p>

INTV_DS_EXT_ID	60	N	A/N	This is the external identifier of the profile to which this data belongs. It is required if profile ID is not specified.
SET_DTTM	26	Y	DateTime	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 follows the seasonal time shifting as defined on the base time zone. Refer to Seasonal Time Shifts for more information. Refer to Algorithms Find the Most Up-To-Date Data for information about why this date/time is important.
SET_STATUS_FLG	2	Y	A/N	Set this value based on whether it will be validated in the system. Set to P for Pending if the profile ID is blank or if the profile's profile type indicates a validation algorithm. Set to C for Complete if the data should not go through any further validation. Refer to Life Cycle of Interval Profile Data for more information

Interval Data

Each piece of interval data for the profile is stored as an Interval Data record linked to the newly created data set. The name of this table is [CI_INTV_DATA](#). The following table describes each column on this table.

Column Name	Length	Req'd	Data Type	Comments
INTV_DATA_SET_ID	12	Y	A/N	This is a foreign key reference to the Interval Data Set that groups the interval values together. This should be your newly created ID from above.
INTV_DATA_DTTM	26	Y	DateTime	This is the date and time of the interval value. This must be stored in standard time for the base time zone. Refer to Time Zone and Time Changes for more information.
INTV_DATA	18.7	Y	N	This is the quantity for the interval.

Interval Data 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_DATA_SET_K](#). The following table describes each column on this table.

Column Name	Length	Req'd	Data Type	Comments

INTV_DATA_SET_ID	12	Y	A/N	This should be your newly created Interval Data Set ID from above.
ENV_ID	6	Y	N	This is the environment id defined on the installation options .

Process IPDSIDB - Determine Profile For Profile Datasets

This process finds pending profile datasets that are not linked to a profile and attempts to determine the appropriate profile by making a match on the external ID.

Note. The status of the profile dataset is set to **Pending** or **Complete** by the system based on whether or not validation algorithms exist for the profile type. Refer to [Life Cycle of Interval Profile Data Sets](#) for more information.

Interval Data Exception

If a validation algorithm or data derivation algorithm finds an error with interval data, a record is written to the interval data exception table with a message indicating the nature of the severe error.

To view the messages associated with the exception records, you must schedule an appropriate To Do entry creation background process. This process generates a To Do entry for every record in the interval data exception table.

The following section describes steps to take to correct the data in error.

How To Correct a Data Set in Error

A data set is in error only if an error is found by a validation algorithm or by a data derivation algorithm. Because there is no limit to the number of validation and data derivation algorithms that you create, there may be just as many ways to resolve these errors. This section describes a typical situation.

Contents

- [Resolving a Derivation Error](#)
- [Resolving a Validation Error](#)

Resolving a Derivation Error

Because the derivation algorithms are creating new data sets, the error is not with the data set being created, but with source data that the algorithm is using to create the new data set.

The following describes the steps involved with fixing an error found by a creation algorithm ([IPFCSACS](#)) supplied with the system.

- Review the To Do entries generated as a result of interval data exceptions.
- The information displayed with the To Do entry provides details of the error. You may also drill down to the Data Set page to view the error.

On the Data Set page, the status is **Error** and the error message is displayed at the bottom. You probably have to fix a source interval data curve. This is common source of errors in *derived* data. One possible error is that data is missing from a source curve.

The following describes the steps to fix a source interval data profile:

- The error indicates that the problem is with the **DEMANDKW** profile relationship type. Use the context menu button adjacent to the owner SA to Go To Service Agreement. You are taken directly to the [Service Agreement Interval Info](#) page. Determine the profile associated with the **DEMANDKW** relationship type.
- Choose **Go To Interval Data** from the context menu for the appropriate Interval Profile ID and find the data set with the missing interval. Verify all the missing intervals that need to be fixed. (Note that the error message only reports the first missing interval it finds. There may be more.)
- Once you have determined the missing intervals, create a new data set for the profile, and enter the missing interval data.

Warning! Existing completed data records should not be changed. If an existing data record is incorrect, you should always add a new data set, with the current date and time, and enter the corrected value for the interval. This is important because the interval data records are the audit records for an existing bill. Refer to [Auditing Your Interval Pricing Bill Lines](#) for more information.

- Once you have filled in the missing intervals, you may navigate back to the [Service Agreement Interval Info](#) page to run the data derivation again. (Note, you may also wait for the [SA Interval Profile Creation](#) background process to run.)
- When the Data Derivation is finished, view the [Interval Data](#) page for the profile with the derived data. You should notice that your **Error** data set is no longer there.

Resolving a Validation Error

The validation algorithms are validating data in existing data sets. The data set in error is marked and the chances are great that the error is with this data.

The following describes the steps involved with fixing an error found by the validation algorithm ([IPFVDS-SMP](#)) supplied with the system.

- Review the To Do entries generated as a result of interval data exceptions.
- The information displayed with the To Do entry provides details of the error. Drill down to the detail of the record on the [Interval Data](#).
- Fix the error and change the status of the data set to **Pending**. The next time the validation algorithm runs, it revalidates this data.

Maintaining Interval Register Data

In this section, we describe information related to maintaining interval register data. This information includes where to maintain the data, the life cycle that this data may go through, uploading data from an external source, and handling exception conditions.

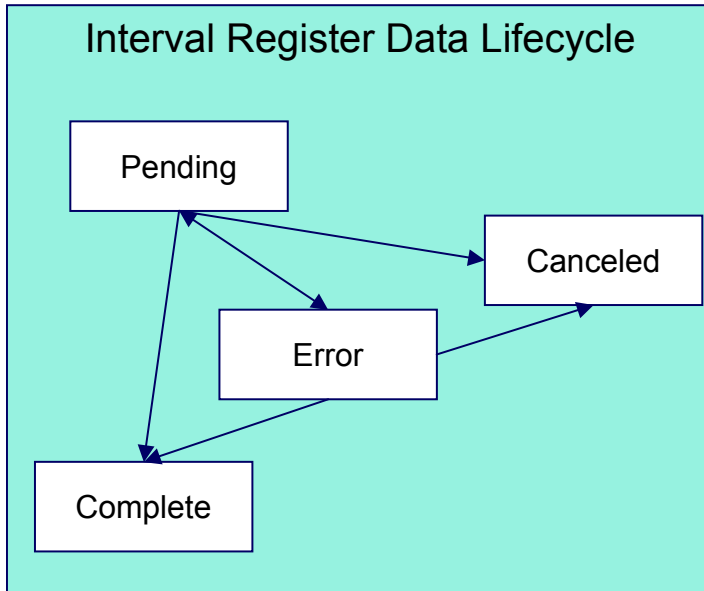
Contents

[Life Cycle of Interval Register Data Sets](#)

[Interval Register Data Maintenance](#)
[Interval Register Data Query](#)
[Upload Interval Register Data](#)
[Process IREGIDB - Determine Register For Register Datasets](#)
[Interval Register Data Exception](#)

Life Cycle of Interval Register Data Sets

The following diagram shows the possible lifecycle of an interval register data set.



Pending

Use this status for data, added to the system, that requires validation to occur before it is available for further processing. Validation is handled through an algorithm linked to the interval register type for this data set's register. Refer to [Setting Up Interval Register Types](#) for more information. When creating a data set, the system automatically sets the status to **Pending** when the interval register type indicates a validation algorithm.

Canceled

A user can cancel a **Pending** or **Error** data set if the data set should not be used for further processing. Refer to [Interval Register Data Maintenance](#) for more information. Additionally, algorithms may be designed to cancel data sets in certain situations.

Error

A validation algorithm may set this status for a data set if it detects an error condition. Data sets in error are also written to the [Interval Register Data Exception](#) table. Refer to [How To Correct a Register Data Set in Error](#). When a validation error has been corrected, change the status of the data set back to **Pending** so it will be reprocessed.

Complete

Only complete data sets are used by derivation algorithms. If data being added to the system has already been validated, it may be uploaded to the system in this status. Validation algorithms may change the status of a data set to **Complete** if no error conditions are detected. When data sets are created, their status is set to **Complete** if the interval register type does not indicate a validation algorithm. For more information about validation algorithms, refer to [Setting Up Interval Register Types](#).

Interval Register Data Maintenance

The Interval Register Data page is used to create and maintain collections of interval register data, grouped together by a data set, for a given interval register.

Open **Main Menu**, **Interval Billing**, **Interval Register Data** to maintain interval register data.

Note. This page is only available if the **Meter Data Management** module is not [turned off](#).

Description of Page

Register Data Set is a concatenation of important information about the data set. The **Register Data Set ID** is a system-assigned random number.

Indicate the **Register ID** to which this data set is linked.

Note. It is possible that the register cannot be identified when adding interval register data. If the **Register ID** is not populated, the **Data Set External ID** must be populated. Refer to [Upload Interval Register Data](#) for more information.

The **Data Set External ID** is the external identifier of the interval register. It will correspond to the channel ID on the interval register. It may be used when a register cannot be found for data interfaced from outside the system. Refer to [Upload Interval Register Data](#) for more information.

The **Set Date/Time** of the Register Data Set represent when this collection of data was entered or received. This is considered a server date and follows the seasonal time shifting as defined on the base time zone. Refer to [Seasonal Time Shifts](#) for more information.

Refer to [Algorithms Find the Most Up-To-Date Data](#) for more information about the Set Date/Time and its use in algorithms.

The **Data Set Type** is available for classification of the data set in some way. The values currently available with the system are:

No Data

Refer to [Fixing Errors with Complete Data](#) for more information about using this data set type.

Estimation

Use this value if the data set contains estimated values based on a validation algorithm. Refer to [IRVA-CLN](#) for an example of an algorithm that creates estimated interval register values.

Note. The values for Data Set Type are customizable using the [lookup table](#). This field name is REG_DS_TYPE_FLG.

The **Status** indicates the status of the interval profile data set. Click **Cancel DS** to cancel a **Pending** or **Error** data set.

Refer to [Lifecycle of Interval Register Data](#) for more information about interval data set status.

Use the **Revert** button to reverse the effect of a **Complete** data set.

Refer to [Fixing Errors with Complete Data](#) 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 Register Date/Time** and the **Interval Register Data** value are captured. This date is considered a logical date and follows the seasonal time shifting as defined on the interval register type. Refer to [Seasonal Time Shifts](#) for more information.

If the data set is in **Error** status, a **Message** that provides the error information is displayed below the interval data collection.

If seasonal time shift records are linked to both the base time zone and the interval register type, the **Seasonal Time Shift** associated with your interval register type is displayed. Refer to [Setting Up Interval Register Types](#) for more information.

If seasonal time shift records are linked to both the base time zone and the interval register type, 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 interval register type or the base time zone do not indicate a seasonal time shift record, the toggle button is not visible and data is displayed and expected in standard time.

Interval Register Data Query

This query shows the most recent interval register data for a register and a time period as of a given time.

Open **Main Menu, Interval Billing, Interval Register Data Query** to display interval register data.

Note. This page is only available if the **Meter Data Management** module is not [turned off](#).

Description of Page

The **Register ID** is the identifier of the register whose data should be 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 are accessed.

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

History

This option displays all interval data for the register and time period, if specified. It includes data in data sets of any status value.

Most Recent Comp. Exc. No Data

This option finds the most recent interval value for each interval from completed data sets only. If the most recent interval is in a **No Data** data set, that interval does not appear.

Most Recent Complete Data

This finds the most recent interval value for each interval from completed data sets only. If the most recent interval is in a **No Data** data set, that interval appears with “No Data” displayed in the **Data Set Type** column.

Refer to [Fixing Errors with Complete Data](#) for more information about **No Data** interval data sets.

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

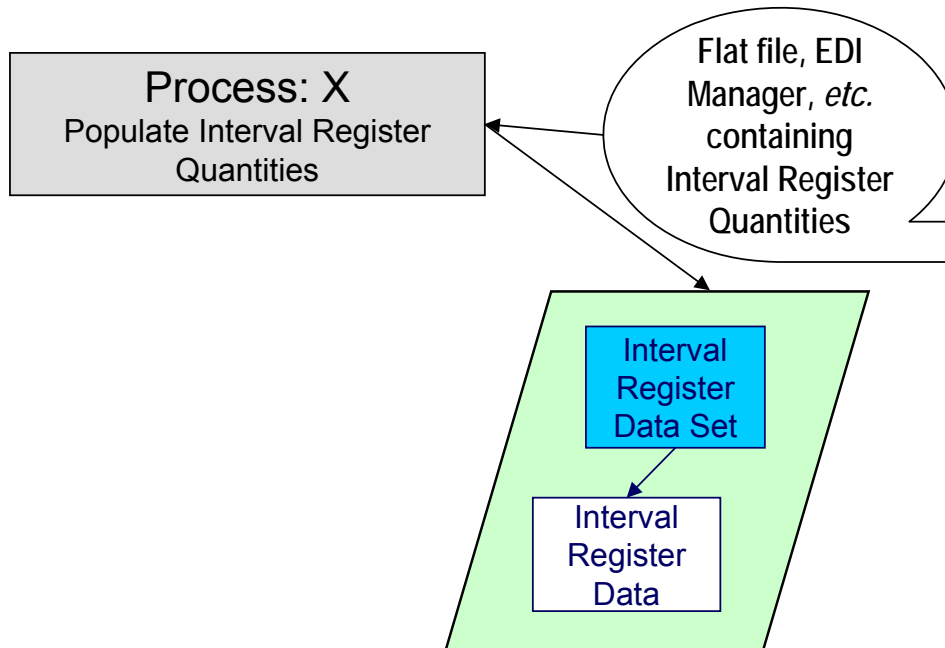
The **Interval Register Date/Time** and **Interval Register Data** value for each interval is displayed. Information about the data set to which this interval is linked is displayed, including the **Data Set Type**, **Set Date/Time** and **Register Data Set ID**.

If seasonal time shift records are linked to both the base time zone and the interval register type, the **Seasonal Time Shift** associated with your interval register type is displayed.

If seasonal time shift records are linked to both the base time zone and the interval register type, 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 interval register type or the base time zone do not indicate a seasonal time shift record, the toggle button is not visible and data is displayed and expected in standard time.

Upload Interval Register Data

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



The topics in this section describe how these processes work.

Contents

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

Process X - Populate Interval Register Data

Process X refers to the mechanism used by your organization to populate the Interval Register Data tables. Process X has the following responsibilities:

- It attempts to determine the appropriate Register to which this data belongs using the channel ID. If a register with this channel ID is not found, the data set is created without being linked to a register and the channel ID is stored as the external ID on the data set.
- It creates a new Register Data Set 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 standard time in the base time zone. Refer to [Time Zone and Time Changes](#) for more information.
- All quantities being uploaded for the same register should be inserted into the interval register data table and should reference the same register data set.
- Set the status of the new data set:
 - If the data set is not linked to a register, set the status to pending.
 - If the register type has a validation algorithm, set the status to pending.
 - If the register type does not have a validation algorithm, this data should be clean. Set the status to complete.
- Inserts a record in the register data set key table.

The topics in this section describe the tables to populate.

Register Data Set

You must create a data set for each batch of records being uploaded in the system for the same register. The name of this table is [CI_REG_DATA_SET](#). The following table describes each column on this table.

Column Name	Length	Req'd	Data Type	Comments
REG_DATA_SET_ID	12	Y	A/N	<p>This is a unique key. This key is referenced on the register data record.</p> <p>It's suggested that this key is built as follows:</p> <ul style="list-style-type: none"> The first five digits are the first five digits of the REG_ID. The last seven digits are randomly generated. <p>This facilitates set based processing of the data. If your process that inserts records on this table is capable of calling a COBOL routine, call CIPZKEYG passing the first five digits and it supplies a seven-digit random number.</p>
REG_ID	10	N	A/N	This is a foreign key reference to the interval register to which this set is linked. It is required if an external ID is not specified.
REG_DS_EXT_ID	60	N	A/N	This is the external identifier of the register to which this data belongs. It should correspond to the channel ID of its register. It is required if register ID is not specified.
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 follows the seasonal time shifting as defined on the base time zone. Refer to Seasonal Time Shifts for more information.</p> <p>Refer to Algorithms Find the Most Up-To-Date Data for information about why this date/time is important.</p>
REG_DS_STATUS_FLG	2	Y	A/N	<p>Set this value based on whether it will be validated in the system.</p> <p>Set to P for Pending if the register ID is blank or if the register's register type indicates a validation algorithm.</p> <p>Set to C for Complete if the data will not go through any further validation.</p>

				Refer to Life Cycle of Interval Register Data for more information
--	--	--	--	--

Register Data

Each piece of interval data for the register will be stored as an Interval Register Data record linked to the newly created data set. The name of this table is [CI_REG_DATA](#). The following table describes each column on this table.

Column Name	Length	Req'd	Data Type	Comments
REG_DATA_SET_ID	12	Y	A/N	This is a foreign key reference to the Register Data Set that groups the interval values together. This should be your newly created ID from above.
REG_DATA_DTTM	26	Y	DateTime	This is the date and time of the register data. This must be stored in standard time for the base time zone. Refer to Time Zone and Time Changes for more information.
INTV_REG_DATA	18.7	Y	N	This is the quantity for the interval.

Register Data 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_REG_DATA_SET_K](#). The following table describes each column on this table.

Column Name	Length	Req'd	Data Type	Comments
REG_DATA_SET_ID	12	Y	A/N	This should be your newly created Register Data Set ID from above.
ENV_ID	6	Y	N	This is the environment id defined on the installation options .

Process IREGIDB - Determine Register For Register Datasets

This process finds pending register datasets that are not linked to a register and attempts to determine the appropriate register by making a match on the external ID of the data set to the channel ID for the register.

Note. The status of the register dataset is set to **Pending** or **Complete** by the system, based on whether or not validation algorithms exist for the register type. Refer to [Life Cycle of Interval Register Data Sets](#) for more information.

Interval Register Data Exception

If a validation algorithm finds an error with interval register data, a record is written to the interval register data exception table with a message indicating the nature of the severe error.

To view the messages associated with the exception records, you must schedule an appropriate To Do entry creation background process. This process generates a To Do entry for every record in the interval register data exception table.

The following section describes steps to take to correct the data in error.

How To Correct a Register Data Set in Error

The validation algorithms are validating data in existing data sets. The following describes the steps involved with fixing an error found by the validation algorithm ([IRGVDS-SMP](#)) supplied with the system.

- Review the To Do entries generated as a result of interval register data exceptions.
- The information displayed with the To Do entry provides details of the error. You may also drill down to the Data Set page to view the error.
- Fix the error and change the status of the data set to **Pending**. The next time the validation algorithm runs, it revalidates this data.

TOU Map Maintenance

The following section describes the pages related to maintaining a TOU map.

Contents

- [TOU Map - Main](#)
- [TOU Map - Related SAs](#)

TOU Map - Main

The TOU Map page is used to create and maintain Time of Use Maps. Open this page using **Main Menu, Interval Billing, TOU Map**.

Note. This page is only available if the **Complex Billing** module is not [turned off](#).

Description of Page

The **TOU Map Info** contains important information about the TOU map. These values only appear after the TOU map exists on the database. The **TOU Map ID** is a system-assigned random number.

Enter the **Description** of the TOU Map.

Indicate the **TOU Map Type** that defines this TOU map.

The **External ID** is available for cross-referencing this TOU map with an external source.

If this map is **SA Owned**, the **SA ID** linked to this map is displayed along with information about the SA.

Use the **Generate** button to generate data for this TOU map based on a TOU map template. Refer to [TOU Map - Generate](#) for more information.

The [tree](#) at the bottom of this page shows information about data sets linked to the TOU map.

- The **Error Data Sets** node displays if any of the data sets linked to the TOU map are in **Error**. Expanding this node displays each error data set.
- The **Data Created Through** node displays the date of the latest complete interval as of the current date. Expanding this node displays the last 10 **Complete** data sets.

TOU Map - Generate

The **TOU Map Template** dialog opens when you click **Generate** on the TOU map page:

Description of Page

Indicate the **TOU Map Template** that you want the system to use as your template for the new data.

The **Set Date/Time** is used for the set date/time on the newly created data set.

Indicate the **Data Set Type**, if applicable.

Indicate the first TOU interval to create by entering the **Start Date/Time** to use for creating the TOU data. Enter either the **End Date/Time** or a value in **Recurring Patterns** to indicate how much data the system should generate.

The holiday fields indicate if and how the system should handle holidays differently from standard days.

Holiday Template

This is the [daily](#) template to use for all holiday days in the period, instead of the TOU map template entered above. Notice that a single [daily](#) Holiday template is used for the period you're generating data for. In other words, the system assumes all holiday days to have the same TOU pattern, i.e. template, throughout the period. If you need to use different TOU patterns for different holiday days you would need to generate data in several steps.

Holiday Shift in Minutes

If the time period covered by a holiday is not 12AM to 12AM, indicate the number of minutes, positive or negative, to shift the holiday. For example, if the holiday is considered 7pm on the eve of the holiday until 7pm on the day of the holiday, enter -300 (i.e. 5 hours) for the shift in minutes.

Calendar

This is the [Calendar](#) where the holidays applicable for this time period are defined.

When you click the **Generate** button, the system creates a new TOU map data set for this TOU map based on the information provided.

The **Seasonal Time Shift** associated with the TOU map type is displayed.

In the **Seasonal Time Shift Remark**, you are reminded that the system expects that the values you enter for the date and time are in local legal time as defined by your TOU map type.

Note. The times on the TOU map components are in legal time. During this generation process the system converts the time definitions into standard time, taking daylight savings into account. Time shifting is based upon the Seasonal Time Shift defined on the template's TOU map type. Refer to [Time Zone and Time Changes](#) for more information.

TOU Map - Related SAs

Open **Main Menu**, **Interval Billing**, **TOU Map** and navigate to the **Related SAs** tab to view service agreements that reference the TOU map.

Note. This page is only available if the **Complex Billing** module is not [turned off](#).

Description of Page

The **TOU Map Info** contains important information about the TOU map. These values only appear after the TOU map exists on the database. The **TOU Map ID** is a system-assigned random number.

Note. If the map is linked to more than 25 service agreements, the search criteria are intentionally left blank in order to avoid retrieving all service agreements (with the resultant slow response times). You must therefore use the **SA Filter** to define the type of service agreements that should be retrieved. See below for more information about this page's search criteria.

Use the **SA Filter** to define the service agreements to 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. You can specify any combination of these fields.
- **All.** Use this option if you do not want to filter service agreements.
- **Person Name.** Use this option to restrict service agreements to those linked to accounts that indicate the given **Name** as the main person.
- **SA Type.** Use this option to restrict service agreements to those linked to a given **CIS Division** and **SA Type**.

Don't forget to click the search button after changing the **SA Filter**.

The grid that follows contains the **Accounts** and **Service Agreements** that match your search criteria.

Maintaining TOU Map Data

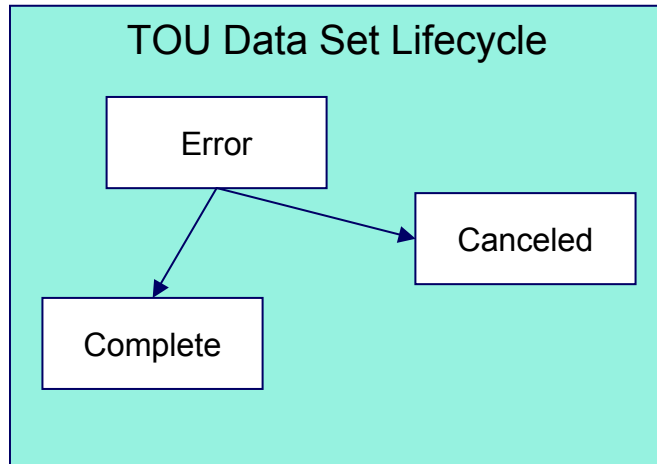
In this section, we describe information related to maintaining time of use data. This information includes where to maintain the data, the life cycle that this data may go through, uploading data from an external source, and handling exception conditions.

Contents

- [Life Cycle of TOU Data Sets](#)
- [TOU Data Maintenance](#)
- [TOU Data Query](#)
- [Upload TOU Map Data](#)
- [TOU Data Exception](#)

Life Cycle of TOU Data Sets

The following diagram shows the possible lifecycle of a TOU data set.



Error

A creation algorithm may set this status for a data set if it detects an error condition. Data sets in this status are also written to the [TOU Data Exception](#) table. Refer to [How To Correct a TOU Data Set in Error](#).

Canceled

A user can cancel an **Error** data set if the data set should not be used for further processing. Refer to [TOU Data Maintenance](#) for more information. Additionally, algorithms may be designed to cancel data sets in certain situations.

Complete

Only complete data sets are used by rate application and derivation algorithms. Data creation algorithms may create a data set in **Complete** status or change the status of a data set to **Complete** if no error conditions are detected. Data sets created online are set to **Complete**.

A user can only change the status of a data set to **Canceled**. All other state transitions are done by an algorithm. For more information about data creation algorithms, refer to [Setting Up TOU Map Types](#).

Note. Creation algorithms may be written to automatically delete any **Error** data sets and start over in an attempt to produce a **Complete** data set. This is similar to the billing process, which deletes error bill segments and tries from the beginning to produce an error free segment. Another alternative is to **Cancel** any data sets in **Error** rather than deleting them.

TOU Data Maintenance

The TOU Data page is used to create and maintain collections of time period definitions, grouped together by a data set, for a given TOU Map. Open **Main Menu, Interval Billing, TOU Data** to maintain TOU data.

Note. This page is only available if the **Complex Billing** module is not [turned off](#).

Description of Page

TOU Data Set is a concatenation of important information about the data set. The **TOU Data Set ID** is a system-assigned random number.

Indicate the **TOU Map ID** to which this data set is linked.

The **Owner SA** indicates the SA that “owns” this data. This information is only populated for **SA owned** TOU maps. This field is useful if the data set you are viewing contains an error and the error is related to a different source data linked to the service agreement. You may use the context button to quickly navigate to the service agreement to view other interval entities related to the service agreement. Refer to [how to correct a TOU data set in error](#) for more information.

The **Set Date/Time** of the TOU Data Set represent when this collection of data was entered or received. This is considered a server date and follows the seasonal time shifting as defined on the base time zone. Refer to [Seasonal Time Shifts](#) for more information.

Refer to [Algorithms Find the Most Up-To-Date Data](#) for more information about the Set Date/Time and its use in algorithms.

The **Data Set Type** is available for classification of the data set in some way. The value currently available with the system is **No Data**. Refer to [Fixing Errors with Complete Data](#) for more information about using this data set type.

Note. The values for Data Set Type are customizable using the [lookup table](#). This field name is TOU_DS_TYPE_FLG.

The **Status** indicates the status of the TOU data set. Click **Cancel DS** to cancel an **Error** data set.

Refer to [Lifecycle of TOU Data](#) for more information about interval data set status.

Use the **Revert** button to reverse the effect of a **Complete** data set.

Refer to [Fixing Errors with Complete Data](#) for more information.

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

At least one TOU data record must exist in the collection of TOU time period definitions. For each TOU data record, the **TOU Data Date/Time** and the **Time of Use** code are captured. This date is considered a logical date and follows the seasonal time shifting as defined on the TOU map type. Refer to [Seasonal Time Shifts](#) for more information.

If the data set is in **Error** status, a **Message** that provides the error information is displayed below the TOU data collection.

If seasonal time shift records are linked to both the base time zone and the TOU map type, the **Seasonal Time Shift** associated with your TOU map type is displayed. Refer to [Setting Up TOU Map Types](#) for more information.

If seasonal time shift records are linked to both the base time zone and the TOU map type, 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 TOU map type or the base time zone do not indicate a seasonal time shift record, the toggle button is not visible and data is displayed and expected in standard time.

TOU Data Query

This query shows the most recent interval data for a TOU Map and a time period as of a given time.

Open **Main Menu, Interval Billing, TOU Data Query** to display TOU map data.

Note. This page is only available if the **Complex Billing** module is not [turned off](#).

Description of Page

The **TOU Map ID** is the identifier of the TOU map whose data should be 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 displays all TOU map data for the TOU map and time period, if specified. It includes data in data sets of any status value.

Most Recent Comp. Exc. No Data

This option finds the most recent interval value for each interval from completed data sets only. If the most recent interval is in a **No Data** data set, that interval does not appear.

Most Recent Complete Data

This finds the most recent interval value for each interval from completed data sets only. If the most recent interval is in a **No Data** data set, that interval appears with “No Data” displayed in the **Data Set Type** column.

Refer to [Fixing Errors with Complete Data](#) for more information about **No Data** data sets.

Once you have selected your criteria, use the search button to display the data. The TOU map data collection contains the following information:

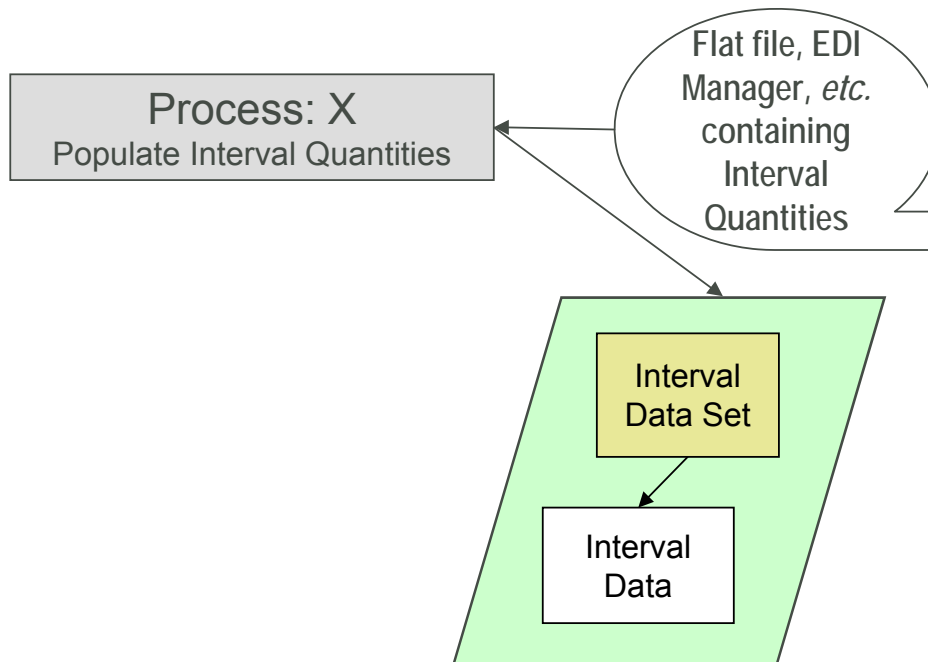
The **TOU Data Date/Time** and **Time of Use** code for each interval is displayed. Information about the data set to which this interval is linked is displayed, including the **Data Set Type**, **Set Date/Time** and **TOU Data Set ID**.

If seasonal time shift records are linked to both the base time zone and the TOU map type, the **Seasonal Time Shift** associated with your TOU map type is displayed.

If seasonal time shift records are linked to both the base time zone and the TOU map type, 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 TOU map type or the base time zone do not indicate a seasonal time shift record, the toggle button is not visible and data is displayed and expected in standard time.

Upload TOU Map Data

The following diagram illustrates the processes involved in the uploading of TOU map data into the system.



The topics in this section describe how these processes work.

Contents

[Process X - Populate TOU Map Data](#)
[TOU Data Set](#)
[TOU Data](#)
[TOU Data Set Key](#)

Process X - Populate TOU Map Data

Process X refers to the mechanism used by your organization to populate the TOU Map Data tables. Process X has the following responsibilities:

- It determines the appropriate TOU map to which this data belongs. (Recall that the TOU map can hold an external ID.)
- It creates a new TOU data set with the current date and time.
- If the TOU map data is from a different time zone or is not in standard time for the base time zone, the time of each interval is converted to standard time in the base time zone. Refer to [Time Zone and Time Changes](#) for more information.

- All quantities being uploaded for the same TOU map should be inserted into the TOU map data table and should reference the same TOU map data set.
- Inserts a record in the TOU data set key table.

The topics in this section describe the tables to populate.

TOU Data Set

You must create a data set for each batch of records being uploaded in the system for the same TOU Map. The name of this table is [CI_TOU_DATA_SET](#). The following table describes each column on this table.

Column Name	Length	Req'd	Data Type	Comments
TOU_DATA_SET_ID	12	Y	A/N	<p>This is a unique key. This key is referenced on the TOU Data record.</p> <p>It's suggested that this key is built as follows:</p> <ul style="list-style-type: none"> • The first five digits are the first five digits of the TOU MAP_ID. • The last seven digits are randomly generated. <p>This facilitates set based processing of the data. If your process that inserts records on this table is capable of calling a COBOL routine, call CIPZKEYG passing the first five digits and it supplies a seven-digit random number.</p>
TOU_MAP_ID	12	Y	A/N	<p>This is a foreign key reference to the TOU map to which this set is linked.</p> <p>Refer to TOU Map Maintenance 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 follows the seasonal time shifting as defined on the base time zone. Refer to Seasonal Time Shifts for more information.</p> <p>Refer to Algorithms Find the Most Up-To-Date Data for information about why this date/time is important.</p>

TOU Data

Each time period definition for the TOU map will be stored as a TOU Data record linked to the newly created data set. The name of this table is [CI_TOU_DATA](#). The following table describes each column on this table.

Column Name	Length	Req'd	Data	Comments
-------------	--------	-------	------	----------

			Type	
TOU_DATA_SET_ID	12	Y	A/N	This is a foreign key reference to the TOU Data Set that groups the TOU definitions together. This should be your newly created ID from above.
TOU_DATA_DTTM	26	Y	DateTime	This is the date and time of the time period definition. This must be stored relative to the base time zone. Refer to Time Zone and Time Changes for more information.
TOU_CD	8	Y	A/N	This is the TOU code that this time period shall be mapped to.

TOU Data 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_TOU_DATA_SET_K](#). The following table describes each column on this table.

Column Name	Length	Req'd	Data Type	Comments
TOU_DATA_SET_ID	12	Y	A/N	This should be your newly created TOU Data Set ID from above.
ENV_ID	6	Y	N	This is the environment id defined on the installation options .

TOU Data Exception

If a derivation algorithm finds an error while creating TOU data, a record is written to the TOU data exception table with a message indicating the nature of the severe error.

To view the messages associated with the exception records, you must schedule an appropriate To Do entry creation background process. This process generates a To Do entry for every record in the TOU data exception table.

The following section describes steps to take to correct the data in error.

How To Correct a TOU Data Set in Error

Because derivation algorithms are creating new data sets, the error is not with the data set being created, but with source data that the algorithm is using to create the new data set.

The following describes the steps involved with fixing an error found by a creation algorithm ([ITMCCOPT](#)) supplied with the system. This algorithm attempts to create TOU map data based on a TOU map template and contract option event information.

- Review the To Do entries generated as a result of TOU data exceptions.
- The information displayed with the To Do entry provides details of the error. Drill down to view more details on the [TOU Data](#) page.

On the TOU data page, the status is **Error** and the error message is displayed at the bottom. In this scenario, the time period covered by the contract event does not overlap with the time period covered by the override TOU map template. This is a shared contract option, but this particular event does not apply to this service agreement. The suggested solution for this case is to cancel the TOU data set in error and to create an override event for the service agreement.

- Cancel the TOU data set record in error.
- Use the context menu button adjacent to the owner SA to **Go To Service Agreement**. You are taken to the interval info tab on the service agreement page.
- Navigate to the [Service Agreement - Contract Option](#) page and add an override for this event.

The next time the creation algorithm runs, it detects the override event and it does not attempt to create TOU data for this scenario.

Contract Option Maintenance

Contract options are used to define special rate options applicable to a customer or a group of customers.

Contents

[Contract Option - Main](#)

[Contract Option - Related SAs](#)

Contract Option - Main

Open **Main Menu**, **Customer Information**, **Contract Option** to maintain contract options data.

Note. This page is only available if the **Complex Billing** module is not [turned off](#).

Description of Page

The **Contract Option Info** displays important information about the contract option. The **Contract Option ID** is a system-assigned random number.

Indicate the **Contract Option Type** for this contract option along with a **Description**.

If this contract option is linked to only one service agreement, indicate the **Create For SA ID**.

The characteristic collection is used to indicate additional information related to your contract option.

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

To modify a characteristic, 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.
-----------------------	--

Characteristic Type	Indicate the type of characteristic.
----------------------------	--------------------------------------

Characteristic Value

Indicate the value of the characteristic.

Contract Option - Related SAs

Open **Main Menu, Customer Information, Contract Option** and navigate to the Related SAs tab to view service agreements that reference the contract option.

Note. This page is only available if the **Complex Billing** module is not [turned off](#).

Description of Page

The **Contract Option Info** displays important information about the contract option. The **Contract Option ID** is a system-assigned random number.

Note. If the contract option is linked to more than 25 service agreements, the search criteria are intentionally left blank in order to avoid retrieving all service agreements (with the resultant slow response times). You must therefore use the **SA Filter** to define the type of service agreements that should be retrieved. See below for more information about this page's search criteria.

Use the **SA Filter** to define the service agreements to 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. You can specify any combination of these fields.
- **All.** Use this option if you do not want to filter service agreements.
- **Person Name.** Use this option to restrict service agreements to those linked to accounts that indicate the given **Name** as the main person.
- **SA Type.** Use this option to restrict service agreements to those linked to a given **CIS Division** and **SA Type**.

Don't forget to click the search button after changing the **SA Filter**.

The grid that follows contains the **Accounts** and **Service Agreements** that match your search criteria.

Maintaining Contract Option Events

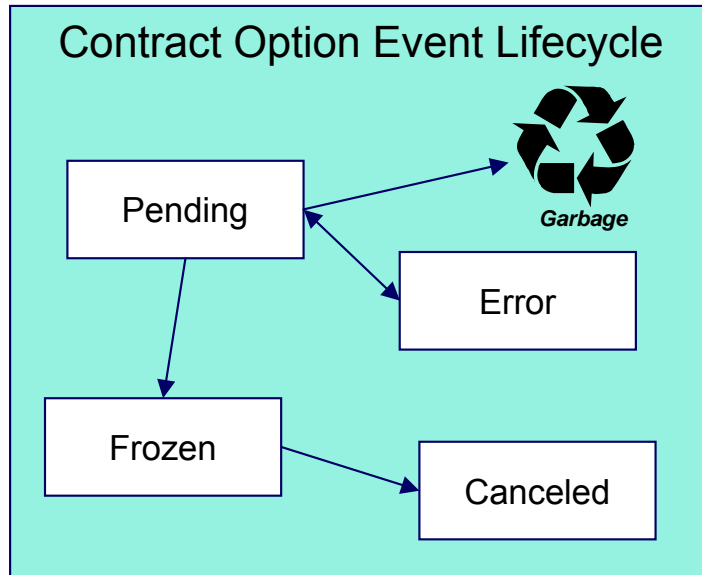
In this section, we describe information related to maintaining contract option events. This information includes where to maintain the data, the life cycle that this data may go through, and handling exception conditions.

Contents

- [Life Cycle of Contract Option Events](#)
- [Contract Option Event Maintenance](#)
- [Contract Option Event Exception](#)

Life Cycle of Contract Option Events

The following diagram shows the possible lifecycle of a contract option event.



Pending

A contract option event may be created in pending status and transition to frozen to be used by processing algorithms. Events in this status may be deleted if they are incorrect.

Error

A pending or frozen validation algorithm defined on the [contract option type](#) may set this status for an event if it detects an error condition. Events in this status also appear on the [Contract Option Event Exception](#) query.

Frozen

Only frozen events are processed by rate algorithms and data derivation algorithms.

Canceled

A frozen event may be canceled if it is incorrect.

Note. Algorithms that process these events need to undo the effect of canceled events.

Contract Option Event Maintenance

Contract option events record the event of a contract option, including its effective period and any characteristic values related to the event. Open **Main Menu, Interval Billing, Contract Option Event** to maintain event data.

Note. This page is only available if the **Complex Billing** module is not [turned off](#).

Description of Page

The **Contract Option Event ID** is a system-assigned random number.

Indicate the **Contract Option ID** to which this event is linked.

Indicate the **Contract Option Event Type** for this event.

The **Start Date/Time** and **End Date/Time** of the contract option event represent when this event is effective. These are considered logical date/times and follow the seasonal time shifting as defined on the contract option type. Refer to [Seasonal Time Shifts](#) for more information.

The **Status** indicates the status of the contract option event. Use the **Freeze** button to change the status of the record from **Pending** to **Frozen**. Use the **Cancel** button to change the status of the record from **Frozen** to **Canceled**.

Note. Any change in status causes the appropriate validation algorithm on the contract option event type, if any, to be executed.

Note. When the pending or frozen validation algorithms find an error, the status of the record is changed to **Error** and an exception record is created. If the cancel validation algorithm finds an error, the error message is displayed to the user. No exception record is created.

Refer to [Lifecycle of Contract Option Events](#) for more information.

The **Status Date/Time** indicates when the record's status changed to its current value. Algorithms that process these events use this information to determine whether the status change affects their processing. For example, if the status is Canceled and the status date/time is more recent than the last time the algorithm processed the events, the algorithm may need to reverse the effect of this event. This date/time is considered a server date and follows the seasonal time shifting as defined on the base time zone. Refer to [Seasonal Time Shifts](#) for more information.

The characteristic collection is used to indicate additional information related to your contract option event.

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

To modify a characteristic, 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:

Characteristic Type	Indicate the type of characteristic.
Characteristic Value	Indicate the value of the characteristic.

Contract Option Event Exception

If a validation algorithm finds an error in contract option event data, a record is written to the contract option event exception table with a message indicating the nature of the severe error.

To view the messages associated with the exception records, you must schedule an appropriate To Do entry creation background process. This process generates a To Do entry for every record in the contract option event exception table.

The following section describes steps to take to correct the data in error.

How To Correct a Contract Option Event in Error

Errors in contract option events are a result of errors detected in a validation algorithm. The validation algorithms provided by the system check for overlap of other events for contract options with the same contract option type.

- Review the To Do entries generated as a result of contract option event exceptions.
- The information displayed with the To Do entry provides details of the error. Drill down to view more details on the [Contract Option Event](#) page.
- Fix the error by resolving the overlap and change the status of the data set to **Pending**. The next time the validation algorithm runs, it revalidates this data.

Interval Billing Calculation Details

When a bill is produced with calculation lines resulting from interval pricing rate components or TOU Pricing rate components, it's likely that a CSR may want to view the calculation detail to verify that this was done correctly.

In order to view the details of the calculation of a bill, the interval pricing rate component and TOU pricing rate components must indicate an audit algorithm. Refer to [Auditing Your Interval Pricing Bill Lines](#) for more information.

Once you have a bill generated, go to **Main Menu, Financial, Bill Segment** and navigate to the **Calc Lines** tab. Locate the lines that were produced as a result of an interval pricing rate component. A drill down button appears in the first column.

When you click the drill down for a given line, you are taken to a page that shows details for the calculation of this line. If your bill line was calculated using an interval pricing rate component, you are taken to the interval pricing details page. If your bill line was calculated using a TOU pricing rate component, you are taken to the TOU pricing details page.

Contents

- [Interval Pricing Details - Main](#)
- [Interval Pricing Details - Details](#)
- [Interval Pricing Details - SA Information](#)
- [TOU Pricing Calc Details - Main](#)
- [TOU Pricing Calc Details - Details](#)
- [TOU Pricing Calc Details - SA Information](#)

Interval Pricing Details - Main

To navigate to this page, you must first go to **Main Menu, Financial, Bill Segment** and display a bill segment with interval data in its calculation lines. From the **Calc Lines** tab, you may choose to audit a bill line that is the result of an interval pricing rate component that has an audit algorithm. Choosing the drill down button for an interval pricing calculation line will bring you to the Interval Pricing - Calc Details - Main page.

Description of Page

Identifiers and descriptions are shown for the **Account, SA** and **Premise** linked to this bill segment.

The **Bill Seg ID**, **Bill Seg Header** and **Calc Line** identify the bill calculation line currently displayed. The **Description on Bill** displays the description of the bill line that is printed on the customer's bill.

The date and time covered by the **Consumption Period** are displayed.

Refer to [Bill Period and Seasonal Time Shifts](#) for more information.

The **Quantity Billed** displays the quantity, followed by a description of the unit of measure, followed by a description of the SQL, if applicable.

The **Total Amount** of the bill line is displayed, followed by the date and time this line was **Calculated On**.

If you click **Rate Info**, a dialog opens displaying information about the rate. The information includes the rate schedule and version, the rate component sequence and description, the interval profile relationship type and bill factor information.

The **Start Date/Time** and **End Date/Time** indicate the range of the data that is currently displayed. To limit the amount of data displayed, enter a different start and end date and time and click **Zoom**. Use **Refresh** to display all the data linked to the bill segment calculation line.

The Interval Billing Graph displays three sets of data and uses color to distinguish between them. **Red** is used for the interval quantities. The Y-axis shows the range of values plotted under the column **UOM**. **Blue** is used for the prices that are coming from the interval billing factor values. The Y-axis shows the range of values plotted under the column **Price**. **Green** is used for the resulting calculation at each interval of quantity multiplied by price. The Y-axis shows the range of values plotted under the column **Cost**. For all three curves, the X-axis shows the date for each interval.

Controlling Which Graphs Are Displayed

All three sets of data are displayed initially, however, you may control which sets of data are displayed by clicking the **UOM**, **Price** or **Cost** buttons to the left of the interval graph. For example to view only the **Price** graph, click the **UOM** and **Cost** buttons.

To redisplay the **UOM** and **Cost** graphs, click the respective buttons again.

Drilling Down In the Graph

While the graph is displayed, you may click on a section of the graph to drill down into more detail. For example, clicking in the middle of the graph will drill down to a small number of days.

You can display hour-by-hour information by clicking further.

At this point, if you click on one of the bars, you will be taken to the appropriate interval data set page to view the interval data itself.

Interval Pricing Details - Details

To navigate to this page, you must first go to **Main Menu**, **Financial**, **Bill Segment** and display a bill segment with interval data in its calculation lines. From the **Calc Lines** tab, you may choose to audit a bill line that is the result of an interval pricing rate component that has an audit algorithm. Once on the **Calc Details** page, you may navigate to the **Details** tab.

Description of Page

The top section of this page is the same as the main page.

Use the **Start Date/Time** and **End Date/Time** to indicate a range of data to display.

The detailed grid displays the **Date/Time** for each interval and the **Quantity**, **Price** and resulting **Amount**. Use the context menu button at the right of the amount column to navigate to either the Interval Value page or the Interval Data page to view the source data.

Below the grid, the **Total Billable SQ** and the **Total Amount** for the bill segment calculation line are displayed.

The **Seasonal Time Shift Remark** indicates whether the data is displayed in legal time or standard time. If the seasonal time shift record on the interval profile type of the first profile matches the seasonal time shift record on the bill factor/characteristic for the price, the intervals are displayed in legal time, according to this seasonal time shift record. Otherwise, the data is displayed in standard time.

Refer to [Seasonal Time Shifts](#) for more information.

Interval Pricing Details - SA Information

To navigate to this page, you must first go to **Main Menu**, **Financial**, **Bill Segment** and display a bill segment with interval data in its calculation lines. From the **Calc Lines** tab, you may choose to audit a bill line that is the result of an interval pricing rate component that has an audit algorithm. Once on the **Calc Details** page, you may navigate to the **SA Information** tab.

Description of Page

The top section of this page is the same as the main page.

The [tree](#) structure on this page shows the profiles found for this service agreement and interval profile relationship type for the time period.

TOU Pricing Calc Details - Main

To navigate to this page, you must first go to **Main Menu**, **Financial**, **Bill Segment** and display a bill segment with TOU pricing data in its calculation lines. From the **Calc Lines** tab, you may choose to audit a bill line that is the result of a TOU pricing rate component that has an audit algorithm. Choosing the drill down button for a TOU pricing calculation line takes you to the TOU Pricing Calc Details - Main page.

Description of Page

Refer to [Interval Pricing Details - Main](#) for information about the information displayed in the top portion of the page. The one difference with the information displayed on this page is that the **Quantity Billed** includes the description of the time-of-use code along with the UOM and SQL.

The TOU Consumption Graph displays in **Red** the interval quantities that were mapped to the time-of-use code represented by the line. The Y-axis shows the range of values plotted under the column **UOM**. The X-axis shows the date for each interval.

Refer to [Interval Pricing Details - Main](#) for information about drilling down into the graph.

TOU Pricing Calc Details - Details

To navigate to this page, you must first go to **Main Menu, Financial, Bill Segment** and display a bill segment with TOU data in its calculation lines. From the **Calc Lines** tab, you may choose to audit a bill line that is the result of a TOU pricing rate component that has an audit algorithm. Once on the **Calc Details** page, you may navigate to the **Details** tab.

Description of Page

The top section of this page is the same as the main page.

Use the **Start Date/Time** and **End Date/Time** to indicate a range of data to display.

The detailed grid displays the **Date/Time** and the **Quantity** for each interval that was mapped to the time of use code associated with this bill line. Use the context menu button at the right of the amount column to navigate to either the Interval Data page or the TOU Data page to view the source data.

The **Seasonal Time Shift Remark** indicates whether the data is displayed in legal time or standard time. If the seasonal time shift record on the interval profile type of the first profile matches the seasonal time shift record on the TOU map type of the first TOU map for the period, the intervals are displayed in legal time, according to this seasonal time shift record. Otherwise, the data is displayed in standard time.

Refer to [Seasonal Time Shifts](#) for more information.

TOU Pricing Calc Details - SA Information

To navigate to this page, you must first go to **Main Menu, Financial, Bill Segment** and display a bill segment with TOU data in its calculation lines. From the **Calc Lines** tab, you may choose to audit a bill line that is the result of a TOU pricing rate component that has an audit algorithm. Once on the **Calc Details** page, you may navigate to the **SA Information** tab.

Description of Page

The top section of this page is the same as the main page.

This page displays two [tree](#) structures. The first tree structure shows the interval profiles found for this SA and interval profile relationship type for this time period. The second tree structure is only visible if the rate component references a TOU map relationship type. It shows the TOU maps found for this SA and TOU map relationship type for this time period.

Interval Billing Examples

Contents

- [How to Link Common Profiles to a Service Agreement](#)
- [How to Link SA Owned Profiles to a Service Agreement](#)
- [Hedge Cover](#)
- [Tunnel Option](#)
- [Override Maps](#)

How to Link Common Profiles to a Service Agreement

The system automatically links **common** profiles to your service agreement if you set up the appropriate start options for your service agreement type. You may also link **common** profiles to your service agreement manually:

- Create the profile (refer to [Interval Profile Maintenance](#) for more information).
- Go to the [Service Agreement - Interval Info](#) tab to link this profile to your service agreement.
- Choose the appropriate profile relationship type for this SA/Profile link. This profile relationship type may be the link to a rate component if this profile is getting billed.
- Indicate the Interval Profile ID that you want to link to the service agreement.
- Save your record.

How to Link SA Owned Profiles to a Service Agreement

Contents

[For The SA That Is The Owner](#)

[For The SA That Is Not The Owner](#)

For The SA That Is The Owner

If this service agreement is the owner SA, the system automatically creates **SA owned** profiles for your service agreement if you set up the appropriate start options for your service agreement type. You may also create **SA owned** profiles for your service agreement manually:

- Create the profile (refer to [Interval Profile Maintenance](#) for more information). Be sure to choose your Service Agreement ID when creating the profile.

Note. If you are on the [Service Agreement Interval Info](#) page, you may navigate to Interval Profile page by choosing Go To Interval Profile +. This brings your service agreement ID to the interval profile page and populates the service agreement field.

- Go to the [Service Agreement - Interval Info](#) tab to link this profile to your service agreement.
- Choose the appropriate Profile Relationship Type for this SA/Profile link. This Profile Relationship Type may be the link to a Rate Component if this profile is getting billed.
- Indicate the Interval Profile ID that you want to link to the service agreement.
- Save your record.

For The SA That Is Not The Owner

This assumes that the profile already exists and is linked to the owner SA. If not, follow the steps above.

- Go to the [Service Agreement - Interval Info](#) tab to link this profile to your service agreement.
- Choose the appropriate Profile Relationship Type for this SA/Profile link. This Profile Relationship Type may be the link to a Rate Component if this profile is getting billed.
- Indicate the Interval Profile ID that you want to link to the service agreement.

- Save your record.

Hedge Cover

Assume your organization has some interval rates with the following conditions. The customer will pay for an agreed upon amount of usage at a fixed price. We will call this amount of usage the hedge cover. It is also known as a “reference curve”. If actual usage falls above the hedge cover, the customer pays the difference at the market price. If the actual usage falls below the hedge cover, the customer receives a credit for the difference at the market price.

In this example, we assume that the actual usage, the hedge cover, the strike price and the market price are all interval curves with the same interval size.

Contents

- [Hedge Cover - Bill Factors](#)
- [Hedge Cover - Profile Relationship Types](#)
- [Hedge Cover - Profile Types](#)
- [Hedge Cover - Rate Schedule and Rate Components](#)
- [Hedge Cover - Common Profile](#)
- [Hedge Cover - SA Type](#)
- [Hedge Cover - Start Option](#)
- [Hedge Cover - Service Agreements](#)
- [Hedge Cover - Typical Business Flow](#)

Hedge Cover - Bill Factors

Strike Price

Set up a bill factor for the Strike Price. On the bill factor indicate that it:

- Has a bill factor type of **Interval**.
- Uses an appropriate characteristic. For the purposes of this example, let's assume that the strike price does not vary by any characteristic and set the value to **N/A**.
- Has a value type of **Unit Rate**.
- Should generate an error if it finds no value.

Set up characteristics for the Strike Price bill factor. On the characteristic indicate:

- Minutes Per Interval of **30** minutes.
- External ID, if applicable, for use by an interface that populates the interval data.

Populate the interval value sets and interval values for the Strike Price bill factor. These values will probably be determined at the time the contract is negotiated. However, they will probably not be entered manually. Rather, your implementers provide an interface that populates this data from an external source. Refer to [Upload Interval Values](#) for information about how this interface should work.

Market Price

Set up a bill factor for the Market Price. On the bill factor indicate that it:

- Has a bill factor type of **Interval**.

- Uses an appropriate characteristic. For the purposes of this example, let's assume that the market price does not vary by any characteristic and set the value to **N/A**.
- Has a value type of **Unit Rate**.
- Should generate an error if it finds no value.

Set up characteristics for the Market Price bill factor. On the characteristic indicate:

- Minutes Per Interval of **30** minutes.
- External ID for use by the interface that populates the interval data.

Set up the interface that periodically populates the interval value sets and interval values for the Market Price bill factor. These values are received from an external source and are uploaded via an interface provided by the implementers. Refer to [Upload Interval Values](#) for information about how this interface should work.

Hedge Cover - Profile Relationship Types

Set up Profile Relationship Types for the following interval profile curves:

- Hedge Cover (**HEDGE**)
- Measured usage (**MEASUSAGE**)
- Difference between Measured and Hedge (**OVRUNDHDG**)

Later we will populate the valid profile types for each relationship type.

Hedge Cover - Profile Types

Create Profile Types

Set up a Profile Type **HEDGE30** for the Hedge Cover, with the following attributes:

- Subtype is **Common**
- Minutes Per Interval is **30** minutes
- Specify an appropriate UOM/SQI

Set up a Profile Type **MEAS30** for the Measured Usage, with the following attributes:

- Subtype is **SA Owned**
- Minutes Per Interval is **30** minutes
- Specify an appropriate UOM/SQI

Set up a Profile Type **OVRUND30** for the Difference between Measured and Hedge, with the following attributes:

- Subtype is **SA Owned**
- Minutes Per Interval is **30** minutes
- Specify an appropriate UOM/SQI

- Profile Algorithm Type is Creation with an appropriate creation algorithm. The algorithm type **IPFCSACS** (supplied with the product) adds or subtracts two curves. The new algorithm should set up the two source Relationship Type parameters as **HEDGE** and **MEASUSAGE** and should indicate “-” for the Add/Subtract parameter.

Update Profile Relationship Types

Now that you have created profile types, go back to each profile relationship type and link the appropriate profile type as valid.

Hedge Cover - Rate Schedule and Rate Components

Create a Rate Schedule for your rate.

For the rate components, let's assume the charges are as follows:

- A bill line with the charges for the hedge cover at the strike price
- A bill line showing the measured usage over the hedge cover at the market price
- A bill line showing the measured usage under the hedge cover at the market price

Note that this rate may contain other charges that are not illustrated here.

The above line items require algorithms to perform the interval pricing calculations. The product is shipped with the algorithm type **RCIPRS**, which applies an interval pricing curve to an interval profile curve assuming that the interval size is the same.

Seq #	RC Type	Description	Profile Relationship Type	Algorithm	Bill Factor
10	Interval Pricing	Hedge cover at the strike price	HEDGE	Algorithm to apply prices to an interval profile curve	Strike Price
20	Interval Pricing	Measured usage over hedge at market price	OVRUNDHDG	Algorithm that applies prices to positive values in an interval profile curve (data does not need to be continuous)	Market Price
30	Interval Pricing	Measured usage under hedge at market price	OVRUNDHDG	Algorithm that applies prices to negative values in an interval profile curve (data does not need to be continuous)	Market Price

Hedge Cover - Common Profile

Create the **common** profile that contains the hedge cover data. On the **common** profile, populate:

- Interval profile type of **HEDGE30**
- External ID, if applicable, for use by an interface that populates the interval data.

Populate the interval profile sets and interval profile data for the hedge cover. These values are probably determined at the time the contract is negotiated. However, they will probably not be entered manually. Rather, your implementers provide an interface that populates this data from an external source. Refer to [Upload Interval Data](#) for information about how this interface should work.

Hedge Cover - SA Type

Determine the SA Type that is used by customers who opt for the hedge cover rate. It may be necessary to set up a new SA type. The SA type chosen must include the following:

- The rate defined above must be linked to the SA type as one of the valid rates.
- The three profile relationship types defined above must be linked to the SA type as valid profile relationship types.

Hedge Cover - Start Option

Create a start option for your SA type. It should set up the hedge cover scenario for a CSR.

- Set the rate for this start option to the hedge cover rate
- Set the default interval profile information as shown in the table below

Profile Relationship Type	Profile Type	Profile
HEDGE		Link the profile created above for the hedge cover
OVRUNDHDG	OVRUND30	
MEASUSAGE	MEAS30	

Hedge Cover - Service Agreements

When starting service for a customer, choose the appropriate SA Type and start option. Upon creation of the SA, the following occurs:

- The appropriate rate is linked.
- The **common** profile created above is linked to the SA.
- Two new **SA owned** profiles are created for the SA using the profile types defined in the start option for measured usage and amount over/under the hedge.

Hedge Cover - Typical Business Flow

Once all the control table data is set and a customer's service agreement has been activated, here is what typically happens in the course of a month:

- Market prices are received periodically and interfaced into the system.
- The customer's measured data is received periodically and interfaced into the system.
- The data derivation background process runs periodically and creates new "over under hedge" data.
- When billing executes, it bills the hedge cover and "over under hedge" data according to the rate algorithms.

Tunnel Option

Let's build on the hedge example. For this example, rather than comparing the measured usage to a single reference curve, the usage will be compared to a range of values at each interval to see if it falls within the range or outside the range. This is sometimes known as a "tunnel" option or a hedge option with "flex" parameters.

Assume that the interval rate for this option has the following conditions.

- For each interval, if the measured usage falls WITHIN the tunnel, the measured amount will be priced at a fixed price.
- For each interval, if the measured usage falls BELOW the LOW tunnel value, the customer will pay the low tunnel value at the fixed price AND will receive a credit for the difference between the low value and the measured amount at a market price.
- For each interval, if the measured usage falls ABOVE the HIGH tunnel value, the customer will pay the high tunnel value at the fixed price AND will pay for the measured amount over the high tunnel value at a market price.

Before we start looking at the data setup, let's talk about how the "tunnel" will be defined. In most cases, each customer does NOT have their own high and low tunnel curves defined. Instead, customer has a "reference curve" (also called "hedge cover") and the high and low values are defined as standard values for your company. These are percentages.

In this example, we assume that the actual usage, the hedge cover, the strike price and the market price are all interval curves with the same interval size.

Contents

- [Tunnel Option - Bill Factors](#)
- [Tunnel Option - Profile Relationship Types](#)
- [Tunnel Option - Profile Types](#)
- [Tunnel Option - Data Derivation Algorithm](#)
- [Tunnel Option - Rate Schedule and Rate Components](#)
- [Tunnel Option - Common Profile](#)
- [Tunnel Option - SA Type](#)
- [Tunnel Option - Start Option](#)
- [Tunnel Option - Service Agreements](#)
- [Tunnel Option - Typical Business Flow](#)

Tunnel Option - Bill Factors

Prices

This example uses the same bill factors for prices as the simple Hedge Cover example. It has a Strike Price bill factor and a Market price bill factor. Refer to [Hedge Cover - Bill Factors](#) for more information.

Tunnel Percentages

We will also use bill factors to define the tunnel percentages. The easiest thing to do is create a separate bill factor for the High Tunnel Percentage and the Low Tunnel Percentage. These will be simple bill factors with a characteristic of *N/A*.

Tunnel Option - Profile Relationship Types

This scenario is similar to the simple Hedge Cover scenario. We still need Profile Relationship Types for the following interval profile curves:

- Hedge Cover (**HEDGE**)
- Measured usage (**MEASUSAGE**)

However, instead of a curve that contains the difference between Measured and Hedge, we need the following profile relationship types:

- Amount of Measured Usage that falls within the Tunnel (**INTUNNEL**)
- Amount of Measured Usage that does not fall within the Tunnel (**OUTTUNNEL**)

Later we will populate the valid profile types for each relationship type.

Tunnel Option - Profile Types

Create Profile Types

This example will use the **HEDGE30** and **MEAS30** Profile Types that were created for the simple Hedge Cover example.

Instead of the profile type for the Difference between Measured and Hedge, this example needs the following profile types.

Set up a Profile Type **INTUN30** for the Amount of Measured Usage that Falls Within the Tunnel, with the following attributes:

- Subtype is **SA Owned**
- Minutes Per Interval is **30** minutes
- Specify an appropriate UOM/SQI
- Profile Algorithm Type is Creation with an appropriate creation algorithm. A new algorithm type will need to be created by the implementers. See [Tunnel Option - Data Derivation Algorithm](#) for information about this algorithm.

Set up a Profile Type **OUTTUN30** for the Amount of Measured Usage that Falls Outside the Tunnel, with the following attributes:

- Subtype is **SA Owned**
- Minutes Per Interval is **30** minutes
- Specify an appropriate UOM/SQI

Note. Although this is a derived curve, it does not have its own data derivation algorithm. Instead, the algorithm linked to the **INTUN30** profile type populates this curve because the calculations for both derived curves can be done with one algorithm.

Update Profile Relationship Types

Now that you have created profile types, go back to each profile relationship type and link the appropriate profile type as valid.

Tunnel Option - Data Derivation Algorithm

It will perform the following functionality:

- It should define two parameters for the source curves, one for the profile relationship type representing the hedge cover (or “reference curve”) and one for the profile relationship type representing the measured usage.
- It should define two parameters for the bill factors for the high and low tunnel percentages.
- It should define a parameter for the profile relationship type for the destination curve of the data that falls outside the tunnel.
- This algorithm retrieves the hedge cover data. For each interval, it calculates the tunnel by applying the high and low tunnel percentages to the hedge cover value for that interval. It then compares the measured data for that interval.
- If the measured data falls within the tunnel, it populates the “Within Tunnel” curve with the measured value. (The “within tunnel” curve is the one with a profile type that contains this algorithm.)
- If the measured data falls below the tunnel, it populates the “Within Tunnel” curve with the low tunnel value and populates the “Outside the Tunnel” curve (based on the parameter passed in to this algorithm) with the difference between the low tunnel value and the measured usage.
- If the measured data falls above the tunnel, it populates the “Within Tunnel” curve with the high tunnel value and populates the “Outside the Tunnel” curve with the difference between the measured usage and the high tunnel value.

Tunnel Option - Rate Schedule and Rate Components

Create a rate schedule for your rate.

For the rate components, let’s assume the charges are as follows:

- A bill line with the charges for the “within the tunnel” data at the strike price
- A bill line with the charges for the “outside the tunnel” data at the market price

Note that this rate may contain other charges that are not illustrated here.

The above line items require algorithms to perform the interval pricing calculations. The product is shipped with the algorithm type [RCIPRS](#), which applies an interval pricing curve to an interval profile curve, assuming that the interval size is the same.

Seq #	RC Type	Description	Profile Relationship Type	Algorithm	Bill Factor
10	Interval Pricing	Within Tunnel data at strike price	INTUNNEL	Algorithm to apply prices to an interval profile curve	Strike Price
20	Interval Pricing	Outside Tunnel data at market price	OUTTUNNEL	Algorithm that applies prices to an interval profile curve (data does not need to be continuous)	Market Price

Tunnel Option - Common Profile

Create the common profile that contains the hedge cover data. If applicable, the curve created in the Hedge Cover example may be reused. Otherwise, refer to [Hedge Cover – Common Profile](#) for information on how to create a common profile.

Tunnel Option - SA Type

Determine the SA type that is used by customers who opt for the tunnel rate. It is possible that the SA type used in the simple Hedge Cover example may be reused. Or it is possible that you want to set up a new SA type.

The SA type chosen must include the following:

- The rate defined above must be linked to the SA type as one of the valid rates.
- The four profile relationship types defined above must be linked to the SA type as valid profile relationship types. (If you reuse the same SA type from the Hedge Cover example, only the two new profile relationship types, **INTUNNEL** and **OUTTUNNEL**, need to be linked at this time.)

Tunnel Option - Start Option

Create a new start option for your SA type that sets up the tunnel option scenario for a CSR.

- Set the rate for this start option to the hedge cover with tunnel option rate.
- Set the default interval profile information as shown in the table below.

Profile Relationship Type	Profile Type	Profile
HEDGE		Link the profile created above for the hedge cover
MEASUSAGE	MEAS30	
INTUNNEL	INTUN30	
OUTTUNNEL	OUTTUN30	

Tunnel Option - Service Agreements

When starting service for a customer, choose the appropriate SA type and start option. Upon creation of the SA, the following occurs:

- The appropriate rate is linked.
- The **common** profile created above is linked to the SA.
- Three new **SA owned** profiles are created for the SA using the profile types defined in the start option for measured usage, inside tunnel and outside tunnel.

Tunnel Option - Typical Business Flow

Once all the control table data is set and a customer's service agreement has been activated, here is what typically happens in the course of a month:

- Market prices are received periodically and interfaced into the system.
- The customer's measured data is received periodically and interfaced into the system.

- The data derivation background process runs periodically and creates new “within tunnel” and “outside tunnel” data.
- When billing executes, it bills the “within tunnel” data and “outside tunnel” data according to the rate algorithms.

Override Maps

Assume that you have contracts with customers where the standard TOU map is overridden on certain days, for example, for interruptions. The customer is informed of these interruptions ahead of time so that they can plan their usage accordingly. Let's assume that the price for usage during these interruption days is more than the standard TOU Price.

Let's assume that the contract identifies the following time period definitions

Time Period Name	TOU Code	Description	Price
On Peak, Summer	ONSUM	The TOU for the standard map for May through September, inclusive for the times 08:00 through 16:59	\$0.0488
Off Peak, Summer	OFFSUM	The TOU for the standard map for May through September, inclusive for the times 00:00 through 07:59 and 17:00 through 23:59	\$0.0391
On Peak, Winter	ONWIN	The TOU for the standard map for October through April, inclusive for the times 08:00 through 16:59	\$0.0421
Off Peak, Winter	OFFWIN	The TOU for the standard map for October through April, inclusive for the times 00:00 through 07:59 and 17:00 through 23:59	\$0.0347
Interruption, On Peak	ONOVR	The TOU for the interruption period. This is not season dependant and covers the times 07:00 through 17:59.	\$0.0504
Interruption, Off Peak	OFFOVR	The TOU for the interruption period. This is not season dependant and covers the times 00:00 through 06:59 and 18:00 through 23:59.	\$0.0418

As you can see, the contract is set up to designate that on these special interruption days, the definition of on peak changes to include two extra hours, and the price per TOU is higher than on the standard days.

Contents

- [Override Maps - TOU Codes and TOU Group](#)
- [Override Maps - Rates and Bill Factors](#)
- [Override Maps - Profile Relationship Type and Profile Type](#)
- [Override Maps - Contract Option Types and Contract Option Event Types](#)
- [Override Maps - Map Relationship Type and Map Type](#)
- [Override Maps - TOU Map Templates](#)
- [Override Maps - SA Characteristic](#)
- [Override Maps - TOU Data Creation Algorithm](#)
- [Override Maps - SA Type](#)
- [Override Maps - Start Option](#)
- [Override Maps - Service Agreements](#)
- [Override Maps - TOU Map Data](#)

Override Maps - Typical Business Flow

Override Maps - TOU Codes and TOU Group

You need to create TOU codes for all the TOUs defined above. Once these are defined, you need to set up a TOU group and link all the above TOUs to this group. For this example, let's call the TOU Group **INTERRUPT**.

Override Maps - Rates and Bill Factors

You need to set up your rate schedule and bill factors needed to handle the TOU pricing for all the TOUs in your TOU group, including the prices for the override TOUs. Refer to [Designing Your Time Of Use Rate Components](#) for more information.

Override Maps - Profile Relationship Type and Profile Type

You need to set up a profile relationship type and a profile type for your measured usage. Let's assume that we have the same control data as that which was used in the hedge cover example.

Override Maps - Contract Option Types and Contract Option Event Types

You need to define a contract option type and event types for the interruption periods.

Create a Contract Option Type for the interruptions. Call it **INTERRUPT**. This contract option type has no special characteristic values.

Create a Contract Option Event Type to define the possible type of interruption events. In this example, let's assume that there is only one type of interruption. Call the event type **INTERRUPT**. This contract option event type has no special characteristic values.

Override Maps - Map Relationship Type and Map Type

You need to define a map relationship type and map type for TOU map.

For this example, create a generic Map Relationship Type of **MEASUSAGE**.

Create a map type that is used for customers using the contract defined above with possible overrides. Call it **INTRPT30**. Define the other attributes as follows:

- TOU Group is **INTERRUPT**
- Minutes Per Interval is **30** minutes
- Algorithm Type is Creation with an appropriate creation algorithm. A new algorithm type needs to be created by the implementers. Refer to [Override Maps - TOU Data Creation Algorithm](#) for information about this algorithm.

Once you have created the map type, go back to the map relationship type and update its valid map types list to include the new map type.

Override Maps - TOU Map Templates

Create a series of TOU map templates for the standard time periods and create a special TOU map template to use for the interruptions.

Override Maps - SA Characteristic

The TOU map creation algorithm needs to know the correct TOU map template to use to generate TOU data when the interruption occurs. Our algorithm assumes that the TOU map template is defined as a characteristic on the service agreement.

Create a foreign key reference for the TOU map template.

Create a new Char type with the following attributes:

- Type is **FK Reference**
- FK reference code is the one created above for TOU map template
- Char Entity is **Service Agreement**

Override Maps - TOU Data Creation Algorithm

The TOU map creation algorithm needs to know the correct TOU map template to use to generate TOU data when the interruption occurs. Our algorithm assumes that the TOU map template is defined as a characteristic on the service agreement.

This algorithm performs the following:

- It should define a parameter of Contract Option Type to indicate the type of contract options whose events should trigger an override interruption TOU map. It should also define a parameter of Char Type that indicates the characteristic type used for TOU map templates.
- It looks for frozen contract option events for the input Contract Option Type effective during the creation time period.
- It also looks for canceled contract option events and attempt to undo their effect.
- It finds a characteristic on the service agreement, whose type matches the input Char Type parameter. It verifies that this is a valid TOU map template.
- It generates a new data set for the TOU map for the time period defined by the contract option events using the TOU map template.

Override Maps - SA Type

Determine the SA type that is used by customers with this option. It may be necessary to set up a new SA type. The SA type chosen must include the following:

- The rate defined above must be linked to the SA type as one of the valid rates.
- The **MEASUSAGE** profile relationship type defined above must be linked to the SA type as a valid profile relationship type.
- The **MEASUSAGE** map relationship type defined above must be linked to the SA type as a valid map relationship type.
- The **INTERRUPT** contract option type defined above must be linked to the SA type as a valid contract option type.

Override Maps - Start Option

Create a start option for your SA type that sets up the override map scenario for a CSR.

- Set the rate for this start option to the appropriate rate.

- In the characteristics collection, indicate the Char Type you defined for TOU map templates and choose the interruptions TOU map template that you created.
- Set the default interval profile, map and contract option information as shown in the tables below:

Profile Relationship Type	Profile Type	Profile
MEASUSAGE	MEAS30	

Map Relationship Type	TOU Map Type	TOU Map
MEASUSAGE	INTRPT30	

Contract Option Type	Contract Option
INTERRUPT	

Note. In this option, there are no **common** profiles or maps.

Override Maps - Service Agreements

When starting service for a customer, choose the appropriate SA type and start option. Upon creation of the SA, the following occurs:

- The appropriate rate is linked.
- The TOU map template for interruptions is linked as a characteristic to the service agreement.
- A new **SA Owned** profile is created for the SA, using the profile type defined in the start option for measured usage.
- A new **SA Owned** TOU map is created for the SA, using the map type defined in the start option.
- A new SA specific contract option is created for the SA, using the contract option type defined in the start option for interruptions.

Override Maps - TOU Map Data

The standard TOU map data must be set up at the beginning of the contract period. A user should go to the [TOU Map Maintenance](#) page and use the **Generate** button to indicate the appropriate TOU map templates to use for the standard data.

Override Maps - Typical Business Flow

Once all the control table data is set and the customer's service agreement has been activated and the standard TOU map data for the customer has been generated, here is what typically happens in the course of a month:

- The customer's measured data is received periodically and interfaced into the system.

- If any special override days occur, a new contract option event is created for the contract option.
- The TOU map creation algorithm runs periodically and it generates new TOU data sets with the TOU Codes and time period definitions as defined above for the interruption days.
- When billing executes, the TOU Mapping, done during rate application, obtains the most up to date TOU map data for each interval. This way, if any special interruption data exists, the measured usage for that day is mapped correctly into the interruption TOUs. Otherwise, the data is mapped to the standard TOUs.
- The correct prices for each time of use period are applied.

Statements

In this section, we describe how to manage your customer's statements.

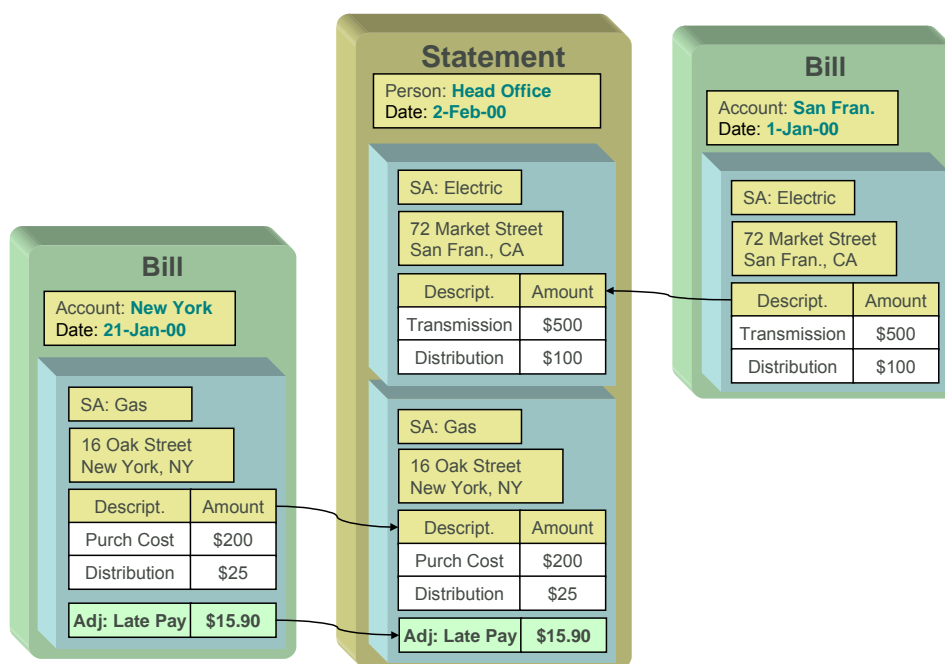
Contents

[The Big Picture of Complex Statements](#)
[Statement Construct Maintenance](#)
[Statement Maintenance](#)

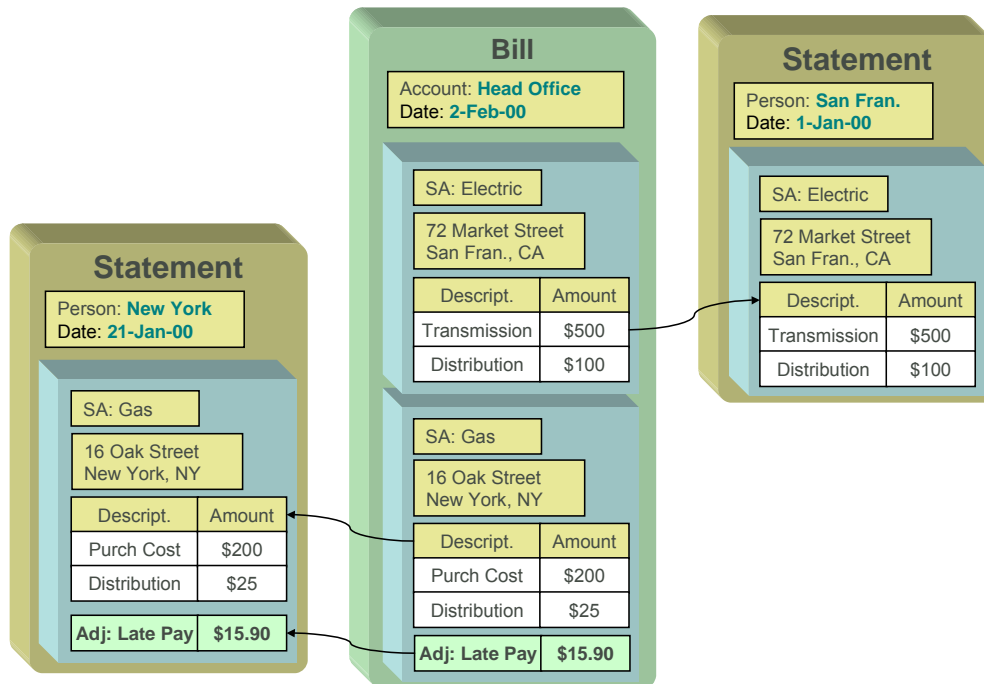
The Big Picture of Complex Statements

Statements allow you to set up a person to receive a consolidated report of the financial activity for one or more accounts and/or service agreements. This allows you to model complex account scenarios, including the following examples.

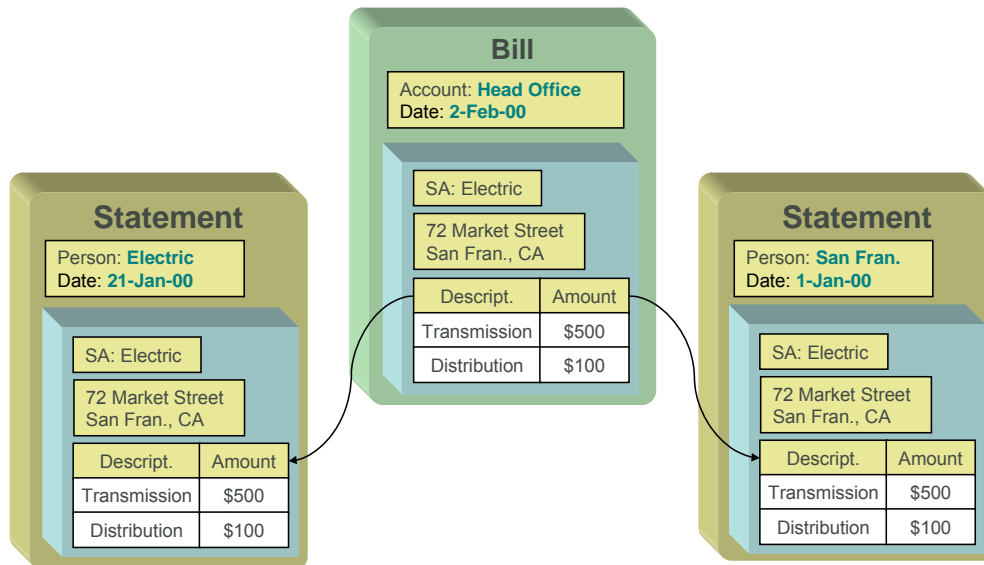
- Satellite offices for a company are each responsible for paying their own bill, but the head office would like a consolidated report of the bills for all the satellite offices once a month. In this case, each satellite office is set up as an account covering the services for their office. The head office does not want to receive individual copies of each bill, but rather a consolidated report of the financial activity for all the offices.



- The head office is responsible for paying the bill. The satellite offices should receive a copy of only their portion of the bill. In this case, a service agreement is set up for each satellite office under the head office's account. Each satellite office should receive a copy of the bill segment information for their office's service agreement.



- The head office is responsible for paying the bill. Multiple departments need to see copies of the electric portion of the bill. As illustrated by this scenario, a service agreement or account may appear on many statements.



The topics in this section provide additional information about how to set up statements.

Contents

[Constructing Statements](#)
[Producing Statements](#)
[Printing Statements](#)

Constructing Statements

Statements are produced for a [person](#). Each statement contains the financial transactions for one or more [service agreements](#). The service agreements may belong to any number of [accounts](#). Over time, a person may receive many statements.

When setting up a person to receive statements, it's important to note that a person can receive different statements for different groupings of service agreements. For example, the head office may want a statement of the financial details for the western division offices separately from the eastern division offices.

For each separate statement a person wishes to receive, you will set up a statement construct. On the statement construct, you define the service agreements and/or accounts whose financial information should appear on the statement. If you specify an account on a statement construct, billing information for all the service agreements linked to the account will be included on the statement. The superset of service agreements and accounts defined on a statement construct are called the "construct details".

Refer to [Statement Construct Maintenance](#) for more information.

Producing Statements

Contents

- [Statement Cycles](#)
- [Create Statements Background Process](#)
- [Statement Routing](#)
- [Where Are Statements Sent](#)
- [On-line Statement Production](#)
- [How to Regenerate a Specific Statement](#)

Statement Cycles

The accounts and service agreements linked to a statement construct may have different bill cycles and as a result, their bills are produced on different dates. So when should a statement containing the consolidation of all this financial information be produced? To answer this question, there is a statement cycle.

Just as with a bill cycle, the statement cycle has a schedule. A statement cycle's schedule controls when a statement should be produced. Each statement construct will point to its own statement cycle.

Refer to [Statement Construct Maintenance](#) and [Statement Cycle](#) for more information.

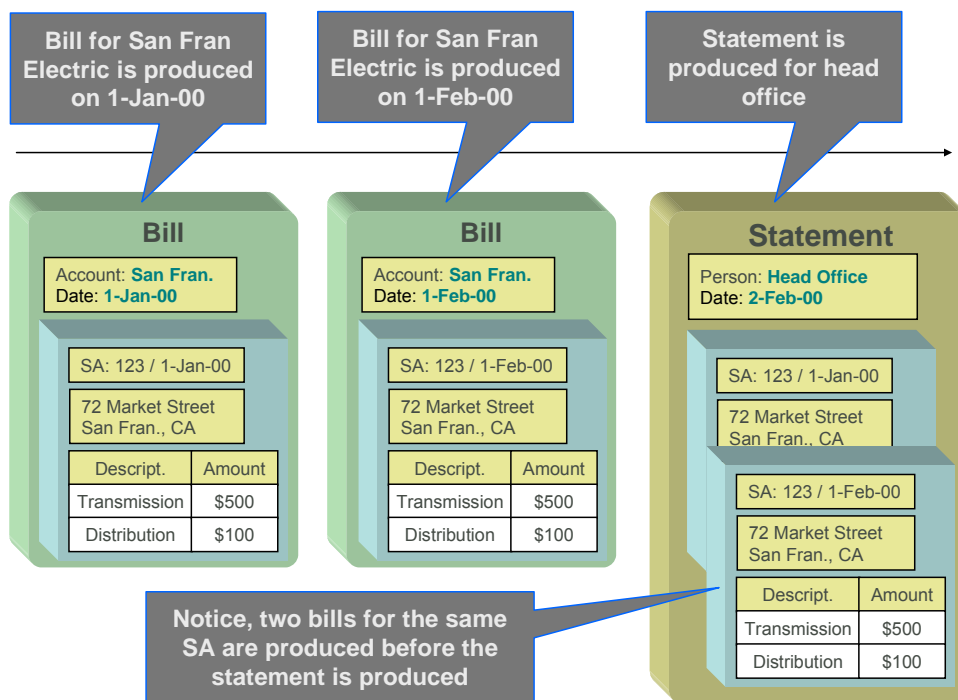
Create Statements Background Process

This statement construction background process (known by the batch control ID of **STMPRD**) is responsible for periodically creating statements. It works as follows:

- All statement cycles are selected with **pending** schedules whose production date is on or before the business date.

- For each such cycle, all of its statement constructs are selected. For each statement construct,
 - If a statement already exists for the cycle schedule and construct, it is deleted (this allows for reproduction after a mess up). Note well, this is true even if the statement is printed as you may have only spotted the problem after viewing the printed statement.
 - A statement is created.
 - All SA's associated with the construct details effective on the business date are extracted.
 - All "SA snapshots" associated with bills whose BILL DATE is within the construct detail's start and end dates that are not already linked to one of the construct's statements are linked to the new statement. Note, an "SA snapshot" is created for every service agreement linked to an account when a bill is completed for an account.
 - If no "SA snapshots" are found, the statement is removed.

Note. The following diagram illustrates how the above logic could result in multiple bills for a single service agreement being linked to the same statement.



Refer to [Statement Maintenance](#) for more information about a statement and its details.

This process is rerunnable. If a problem occurred during the statement creation, simply run the background process again. It will clean up all existing statements for each construct in the current cycle schedule and start again.

Refer to [Printing Statements](#) for more information for how a separate background process uses the information snapshot on a statement to construct the extract file used to print the statements.

Statement Routing

Just as with bills, statements require routing information in order for the system to understand how and where to send the statements. For example, the system needs to know the method of routing, the number of copies and the format of the statement. This information is captured on a statement construct.

Where Are Statements Sent

The system extracts a statement's address as per its statement construct's address source. Note, if the address source is **Person** and the person has a [seasonal address](#) effective on the business date, the seasonal address will be used.

If you need to reprint a statement because it was addressed incorrectly, correct the person's address, and then reprint the statement (by pressing the **Reprint** button on the statement). When you press this button, the statement will be marked for downloading. When your download process next runs, it simply extracts the statement construct's current address.

On-line Statement Production

Statements may be produced on-line as well as in batch. To produce a statement on-line, simply go to the Statement page, select the correct Construct ID and choose the Generate button.

This statement will include the financial information related to bills for the accounts and service agreements linked to this construct, with the following criteria:

- The bills do not already appear on a statement
- The bill dates fall within the "effective period" for the statement construct. In other words, the bill falls within the start and end period for the statement construct details.

Refer to [Statement Maintenance](#) for more information.

How to Regenerate a Specific Statement

There may be times when a specific statement needs to be regenerated. For example, perhaps a cancel / rebill has occurred since the statement was produced, and you wish to recreate the statement to reflect this.

If you need to regenerate a statement using up-to-date information, simply delete the statement and generate a new statement. This may be done on-line.

Note. Statements may be deleted, even if they have already been printed. This enables you to delete and regenerate statements.

Refer to [Statement Maintenance](#) for more information.

Printing Statements

The contents of this section describe the technical implementation of printing statements.

Contents

- [Statement Route Types Control The Information Merged Onto Statements](#)
- [Technical Implementation Of Online Statement Image](#)
- [Technical Implementation Of Printing Statements In Batch](#)
- [Reproducing The Statement Print Flat File](#)
- [How To Reprint A Specific Statement](#)
- [The Doc 1 Records](#)

Statement Route Types Control The Information Merged Onto Statements

Every statement references a [statement route type](#). The route type controls the following statement print functions:

- It contains an algorithm that is responsible for extracting the information merged onto your statements. Specifically, algorithms of this type create the “flat file records” that are passed to your statement 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 statement print software calls these algorithms to construct the “flat file records” for each statement.
 - If your statement print software has the ability to construct a real-time image of a statement (in a PDF), you can plug-in an **Online Statement Image** algorithm on the [Installation Record](#). This type of algorithm will call the statement’s dispatch group extract algorithm to extract the information that is merged onto the statement. Refer to [Technical Implementation Of Online Statement 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 statement print software. The base package example of this process (known by the batch control ID of [STMDWLD](#)) simply calls each statement’s construct’s route type’s extract algorithm to format the information placed onto the flat file.

Technical Implementation Of Online Statement Image

Users can view an image of any statement if you setup the following:

- Plug-in an **Online Statement Image** construction algorithm on the [Installation Record](#). Refer to [ONSD-ST](#) for an example of such an algorithm. Note, if your statement print software is not capable of producing a PDF containing an image of a statement, users will not be able to view images of statements.
- Plug-in the appropriate extract algorithm on each [statement route type](#). Algorithms of this type format the records that contain the information that is passed to your printing software. Refer to [STEX-TX](#) for an example of such an algorithm.

When you plug-in these algorithms, a button appears on [Statement - Main](#). When a user presses this button, the following takes place:

- The installation record’s **Online Statement Image** construction algorithm is executed.

- This algorithm calls the statement's route type's extract algorithm. This algorithm constructs the information that's merged onto the statement and returns it to the **Online Statement Image** algorithm. This algorithm, in turn, passes it to your statement print software.
- Your statement print software renders an image of the statement in a PDF and returns it to the **Online Statement Image** algorithm.
- And finally, the **Online Statement Image** algorithm displays the PDF in a separate Adobe session.

Technical Implementation Of Printing Statements In Batch

The batch process that extracts statement information (known by the batch control ID of [STMDWLD](#)) reads all statements in a given run number that are marked with its batch control ID. For each statement, it creates numerous records on a flat file. These records contain the information that is merged onto your statements.

The base package example of this process (known by the batch control ID of [STMDWLD](#)) simply calls the extract algorithm on the statement route type to format the information placed onto the flat file. Refer to [Statement Route Types Control The Information On Statements](#) for more information.

If your software doesn't support online statement images. The algorithm that formats statement extract records that's plugged in on [statement route type](#) serves two purposes: 1) it formats the records used to construct online images of a statement, and 2) it formats the records downloaded to your statement print software in batch. If your statement print software does not support the rendering of statement 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 statement route type).

Reproducing The Statement Print Flat File

You can reproduce the flat file at any time. Simply request the [STMDWLD](#) process and specify the run number associated with the historic run.

How To Reprint A Specific Statement

If you need to reprint a specific statement, navigate to [Statement – Main](#) and press the **Reprint** button.

If your implementation has [enabled the online creation of statement images](#), you can also press the **Display Statement** button on this page and then print the resultant PDF on your local printer.

The Doc 1 Records

Numerous different types of records are interfaced to the Doc 1 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 statement template.
- The next 186 bytes are called the **Sort Key**. This field is used to ensure the various records used to construct a printed statement 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 statements are printed.
- The remaining bytes contain the **Statement Information** that appears on the printed statement. The type of information differs for each type of record.

The topics in this section describe each component.

Contents

[Printer Keys \(Record Types\)](#)

[Sort Key](#)

[Mailing Key](#)

[Statement Information](#)

Printer Keys (Record Types)

The following table defines each printer key (i.e., record type) on the statement print flat file.

Record Type	Description	Notes
0010	Global extract information	Required. 1 per flat file.
0100	Statement record	Required. 1 per statement.
0120	Total by currency record	Required. At least one per statement.
0500	Construct detail record	Required. At least one per construct detail linked to the statement construct.
0900	Service agreement record	1 per SA
1100	Bill record	1 per bill
1300	Financial Transaction (FT) record	1 per FT linked to the SA.
3600	End of record type 1100	1 for each bill
3700	End of record type 0900	1 for each SA
5000	End of record type 0500	1 for each statement construct detail
9999	End of record type 0100	1 for each Statement

Sort Key

The following table defines the structure of the sort key that appears on each statement print record. Please note that different components are filled in for each printer key.

Field Name	Format	Record Type	Source/Value/Description
Statement ID	A12	All	Prime key of the statement
Sequence Number	N2	All	Set to "1"
Copy Number	N2	All	Defaulted to "1" on the first copy, set to "1" when a statement is requested on-line
Statement record group	A2	0010	Blank

		0100	10
		0120	20
		0500, 0900, 1100, 1300, 3600, 3700, 5000	30
		9999	60
Construct detail sort sequence 1	N3	0010, 0100, 0120, 9999	blank
		0500, 0900, 1100, 1300, 3600, 3700, 5000	Statement construct's PRT_SEQ
Construct detail sort sequence 2	A10	0010, 0100, 0120, 9999	blank
		0500, 0900, 1100, 1300, 3600, 3700, 5000	Statement construct's STM_CNST_DTL_ID
Premise Indicator	A1	0010, 0100, 0120, 0500, 9999	Blank
		5000	'Z'
		0900, 1100, 1300, 3600, 3700	For a non-premise, this will be 'N' otherwise 'Y'
Premise State	A6	0010, 0100, 0120, 0500, 5000, 9999	Blank
		0900, 1100, 1300, 3600, 3700	Premise's state (if this is for a non-premise, this will be blank)
Premise City	A30	0010, 0100, 0120, 0500, 5000, 9999	Blank
		0900, 1100, 1300, 3600, 3700,	Premise's city (if this is for a non-premise, this will be blank)
Premise Address 1	A58	0010, 0100, 0120, 0500, 5000, 9999	Blank
		0900, 1100, 1300, 3600, 3700	Premise's address line 1(if this is for a non-premise, this will be blank)
Premise Id	A10	0010, 0100, 0120, 0500, 5000, 9999	Blank
		0900, 1100, 1300, 3600, 3700	Premise's ID (if this is for a non-premise, this will be blank)
SA Print Priority	A2	0010, 0100, 0120, 0500, 5000, 9999	Blank
		0900, 1100, 1300, 3600, 3700	The SA's SA type's print priority
SA Id	A10	0010, 0100, 0120, 0500, 5000, 9999	Blank
		0900, 1100, 1300,	The SA's SA ID

		3600, 3700	
SA record group	A2	0010, 0100, 0120, 0500, 5000, 9999	Blank
		0900	10
		1100, 1300, 3600	20
		3700	30
Bill Date	A10	0010, 0100, 0120, 0500, 0900, 3700, 5000, 9999	Blank
		1100, 1300, 3600	Bill's bill date in the format YYYYMMDD.
Bill Id	A12	0010, 0100, 0120, 0500, 0900, 3700, 5000, 9999	Blank
		1100, 1300, 3600	Bill's bill id
Bill record group	A2	0010, 0100, 0120, 0500, 0900, 3700, 5000, 9999	Blank
		1100	10
		1300	20
		3600	30

Mailing Key

The following table defines the structure of the mailing key that appears on each statement print record.

Field Name	Format	Source/Value/Description
Postal code	A12	This is the postal code on address associated with the statement.

Statement Information

The topics in this section describe the information that appears on each on each statement print record.

Contents

[Global Extract Information Record \(0010\)](#)
[Statement Record \(0100\)](#)
[Total By Currency Record \(0120\)](#)
[Statement Construct Detail Record \(0500\)](#)
[Service Agreement Record \(0900\)](#)
[Bill Record \(1100\)](#)
[Financial Transaction \(FT\) Record \(1300\)](#)
[End Bill Record \(3600\)](#)
[End Service Agreement Record \(3700\)](#)
[End Statement Construct Detail Record \(5000\)](#)
[End Statement Record \(9999\)](#)

Address Sub-record

Global Extract Information Record (0010)

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.
STATEMENT_CNT	N10	Count of statements on this extract. Multiple copies of a statement are counted multiple as well.
EXTRACT_DTTM	A31	System time of extraction. Formatted according to user profile.
STMT_ID_START	A12	This is only used when a statement is produced as a result of an online request (from Statement – Main). It contains the Statement Id to extract.
STMT_ID_END	A12	This is only used when a statement is produced as a result of an online request (from Statement – Main). It contains the Statement Id to extract.
THREAD_NBR	N10	This field is only used if statements are extracted in batch. It contains the thread number in which the statement was extracted.
THREAD_CNT	N10	This field is only used if statements are extracted in batch. It contains the total number of threads that were extracted.
STATEMENT_IND	A1	Defaulted to "Y"

Statement Record (0100)

Field Name	Format	Source/Value/Description
STMT_ID	A12	CI_STM
PER_ID	A10	CI_STM_CNST
STMT_DT	A31	STM_DT from CI_STM. Formatted according to user profile.
COPY_NBR	N1	"1"
NBR_STMT_COPIES	N1	"1"
ENTITY_NAME	A64	Main name of CI_PER (from CI_PER_NAME).
ADDRESS_SBR	A361	Address where the statement should be sent (the address is defined on the statement's statement construct).
ENTITY_NAME1	A64	CI_PER
ENTITY_NAME2	A64	CI_PER
ENTITY_NAME3	A64	CI_PER
DESCR50	A50	CI_STM_CNST

Total By Currency Record (0120)

One total record will be created for each currency code associated with the FT's associated with the SA snapshots linked to the statement.

Field Name	Format	Source/Value/Description
CURRENCY_CD	A3	This is the currency code associated with the summary information.
CUR_BAL	A30	This is the sum of CUR_AMT snapshot on the SA snapshots

Field Name	Format	Source/Value/Description
		(CI_BILL_SA) linked to the statement. SA snapshots (CI_BILL_SAs) are referenced on CI_STM_DTL.
BILL_AMT	A30	This is the sum of CUR_AMT from CI_FT's classified as "corrections" and "current charges" on bill's associated with the statement. Please note, only those FT's associated with SAs linked to the statement will be amalgamated. This is formatted according to user's display profile.
CUR_ADJ_AMT	A30	This is the sum of CUR_AMT from CI_FT's classified as "adjustments" on bills associated with the statement. Please note, only those FT's associated with SAs linked to the statement will be amalgamated. This is formatted according to user's display profile.
CUR_PAY_AMT	A30	This is the sum of CUR_AMT from CI_FT's classified as "payments" on bills associated with the statement. Please note, only those FT's associated with SAs linked to the statement will be amalgamated. This is formatted according to user's display profile.

Statement Construct Detail Record (0500)

Field Name	Format	Source/Value/Description
STM_CNST_ID	A12	CI_STM_DTL
SA / Account Indicator	A1	A switch to indicate if the statement construct is for a service agreement of an account. 'S' for service agreement, 'A' for account.
Unique ID	A10	The unique identifier of the account or service agreement
SA / Account Info	A160	Contains the standard information about the account or service agreement
STMT_PRT_DESCR	A50	CI_STM_CNST

Service Agreement Record (0900)

Field Name	Format	Source/Value/Description
SA_ID	A10	CI_SA
Address Information	A120	This is the standard format of a premise address that is displayed throughout the system.
SA Information	A120	This is the standard SA information that is displayed throughout the system
SVC_TYPE_CD	A2	CI_SA_TYPE

Bill Record (1100)

Field Name	Format	Source/Value/Description
BILL_ID	A12	
Completion Date	A31	Date formatted according to user profile.
Due Date	A31	Date formatted according to user profile.
SA Ending Balance	A30	
FT Exist Indicator	A1	

Financial Transaction (FT) Record (1300)

Field Name	Format	Source/Value/Description
FT_ID	A12	CI_FT
ARS_DT	A31	CI_FT (The date when the transaction starts aging). Date formatted according to user profile.
AMOUNT	A30	CUR_AMT from CI_FT formatted according to user display profile.
FT_DESCR	A120	Formatted FT information.

End Bill Record (3600)

Field Name	Format	Source/Value/Description
Dummy field		

End Service Agreement Record (3700)

Field Name	Format	Source/Value/Description
Dummy field		

End Statement Construct Detail Record (5000)

Field Name	Format	Source/Value/Description
Dummy field		

End Statement Record (9999)

Field Name	Format	Source/Value/Description
Dummy field		

Address Sub-record

The address sub-record that in statement record (0100) is composed of the following:

Field Name	Format	Description
COUNTRY	A3	
ADDRESS1	A64	
ADDRESS2	A64	
ADDRESS3	A64	
ADDRESS4	A64	
CITY	A30	
NUM1	A6	
NUM2	A4	
HOUSE_TYPE	A2	
COUNTY	A30	
STATE	A6	
POSTAL	A12	

Field Name	Format	Description
GEO_CODE	A11	
IN_CITY_LIMIT	A1	

Statement Construct Maintenance

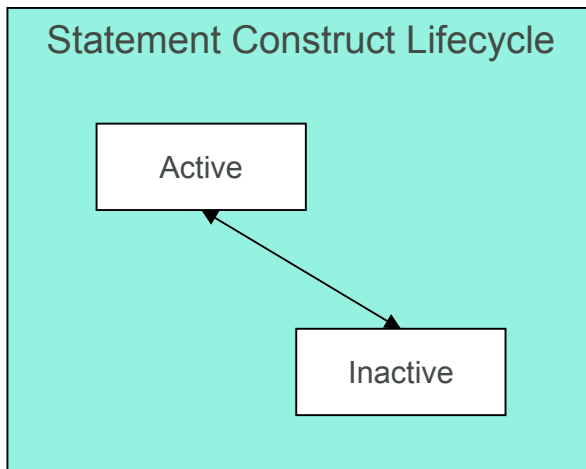
The statement construct enables you to indicate the collection of accounts and/or service agreements whose financial details should be consolidated for a statement to a given person. The statement construct also contains routing information needed by the system to send the statement to the person in the preferred manner.

Contents

- [Lifecycle of a Statement Construct](#)
- [Statement Construct - Main](#)
- [Statement Construct - Details](#)

Lifecycle of a Statement Construct

The following diagram illustrates the lifecycle of a statement construct.



Active

The statement construct will be in this status while a person is actively receiving statements for the accounts and service agreements linked to this construct. A statement construct may transition from this status to **inactive**.

Inactive

The statement construct will be in this status when a person no longer wishes to receive statements for the accounts and service agreements linked to this construct. A statement construct may transition from this status back to **active**.

Statement Construct - Main

Open this page using **Customer Information, Statement Construct**.

Description of Page

Statement Construct is a concatenation of summary information about this record. It is composed of the name of the statement person, the print description and the status.

Statement Construct ID is the unique system-generated identifier for this statement construct.

Person ID is the person who receives the future statements.

Note. A person may be linked to many statement construct records.

Status indicates whether or not this person is currently receiving statements. The values are **Active** and **Inactive**.

Refer to [Lifecycle of a Statement Construct](#) for more information.

Address Source indicates where the source of the statement's address. The values are **Person** and **Mailing Premise**. If the value is person, the person's [mailing address](#) is used. If the value is mailing premise, indicate the **Mailing Premise** to use.

Statement Cycle controls when statements are produced for this statement construct. Refer to [Statement Cycle](#) for a description of how the production schedule is maintained.

Statement Route Type controls how the statement is routed to the **Person** (e.g., via email, postal service, EDI, etc.). Refer to [Statement Route Type](#) for more information on setting up statement route types.

Number of Copies indicates how many copies of the statement the person wishes to receive.

Statement Format indicates if the person receives a **Detailed** or a **Summary** statement.

Note. The values for this field are customizable using the Lookup table. The values need to match the formats supported by your statement print software. This field name is STM_FORMAT_FLG.

Print Description is a brief description that can be printed on the statement.

The tree shows the statement construct's details. The nodes expand to show account, premise and service agreement information for the construct details.

Large statement constructs. The tree is hard to use if the statement construct has many details. In this situation, transfer to the adjacent tab page and use the filter to restrict the details that are shown.

Statement Construct - Details

This page is used to maintain the accounts and service agreements where financial transactions should appear on statements produced using this statement construct. Open this page using **Customer Information, Statement Construct, Details**.

Description of Page

Statement Construct is a concatenation of summary information about this record. It is composed of the name of the statement person, the print description and the status.

Statement Construct ID is the unique system-generated identifier for this statement construct.

If a statement construct has a large number of details, you can use the **Details Filter** to limit the details that appear in the grid. The following options are available:

- **All.** Use this option to view all details linked to the statement construct.
- **Person Name.** Use this option to restrict details linked to accounts whose main customer has a primary name that matches an input **Name**.
- **SA's at an Address.** Use this option to only show details linked to service agreements that are linked to service points associated with a given **Address, City** and/or **Postal** code. Note, you can specify any combination of these fields.
- **Statement Print Description.** Use this option to restrict details to those with a given **Description**.

Don't forget to click the search button after changing the filter.

The grid contains the Accounts and/or Service Agreements whose financial information contributes to this statement construct. Each record in the grid contains the following information:

- **Construct Detail Type** defines if the construct detail is for an entire **Account** or a single **Service Agreement**. This field is protected if a statement detail was produced using this construct.
- **Service Agreement/Account** identifies the account (for **Account** construct details) or the service agreement (for **Service Agreement**) construct details. This field is protected if a statement detail was produced using this construct.
- **Statement Print Description** is available for use by the extract program to include on the printed statements to help the recipient identify the details.
- **Print Order** control where the financial transactions associated with the account / service agreement appear on the printed statement.
- **Start Date** is the date that this account / service agreement's financial transactions should begin appearing on statements.
- **End Date** is the date that this account / service agreement's financial transactions should stop appearing on statements.

Note. Only financial transactions linked to bills whose **BILL DATE** is within the construct detail's **Start Date** and **End Dates** will appear on statements.

- **Construct Detail ID** is a system generated unique identifier of the construct detail.

Statement Maintenance

Statements allow you to set up a Person to receive a consolidated report of the financial activity for one or more accounts and/or service agreements. The topics in this section describe how to maintain statements.

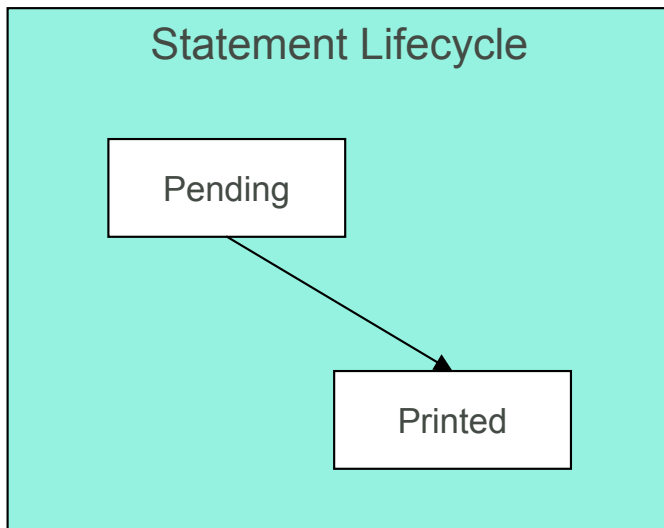
The system creates most statements behind-the-scenes. Most statements are created by the system when it processes the statement cycle schedule. You should only have to access the statement pages to regenerate a statement or to add a statement real-time. For information about how the system creates statements, refer to [The Statement Creation Background Process](#).

Contents

- [Lifecycle of a Statement](#)
- [Statement - Main](#)
- [Statement - Details](#)

Lifecycle of a Statement

The following diagram illustrates the lifecycle of a statement.



Pending

A statement is created in this state and will remain in this state until the statement details are extracted. When that occurs, it will transition to ***printed***.

Printed

A statement transitions to this state after the details of a statement have been extracted.

Statement - Main

Open this page using **Financial, Statement**.

Correcting erroneous statements. It's important to be aware that there are very few fields that are directly modifiable by a user. This is because a statement is a summary of financial transactions. If you need to change information on a statement, you must change the source information (e.g., change the statement's construct or complete additional bills) and then regenerate the statement. Refer to [How To Regenerate A Statement](#) for more information.

Description of Page

Statement is a concatenation of summary information about this record. It is composed of the name of the statement person, the statement construct print description, the create date of the statement and the statement status.

Statement ID is the unique system-generated identifier for this statement.

Indicate the **Construct ID** for the statement construct whose construct details should be included in this statement.

Address is a display-only field that contains the address to which the statement will be sent.

The **Statement Status** indicates the current status of the statement. The values are *Pending*, *Printed*.

Refer to [Lifecycle of a Statement](#) for more information.

The **Create Date** indicates the date that this statement was created.

The **Batch Control** and **Batch Number** are the batch process and run in which the statement was (or will be) sent to the recipient.

Refer to [Printing Statements](#) and to [Statement Route Type](#) for more information.

The **Statement Action** area contains buttons that you use to generate, reprint or delete a statement.

- The **Generate** button enables you to create a statement on-line. Refer to [On-line Statement Production](#) for more information.
- The **Reprint** button will stamp the latest run number for the statement's batch control. This will enable the statement to be reprinted the next time the batch control process executes. Refer to [How to Reprint A Specific Statement](#) for more information.
- The **Delete** button will remove the statement and its details from the database. Refer to [How to Regenerate a Specific Statement](#) for more information.
- The **Display** button will display an online image of the statement when pressed. Refer to [Online Statement Image](#) for more information.

Note. You can only use the **Display** button if your system has been configured to display an on-line image; otherwise, a message indicating that the service is not available will appear. This option can only be configured by your technical staff. Refer to [Technical Implementation Of Online Statement Image](#) for more information.

The bottom portion of this page is dedicated to a [tree](#) that shows the statement details. The nodes expand to show every service agreement that will appear on the statement and the financial details that will be swept onto the statement for each service agreement. You may transfer to the details of any of the nodes by selecting that node. Context menus are also available for various nodes to facilitate easy navigation to other pages.

Large statements. The tree will be hard to use if the statement has many service agreements that contributed financial transactions to the statement. In this situation, transfer to the adjacent tab page and use the filter to restrict the service agreements that are shown.

Statement - Details

This page is used to view the service agreements that contributed financial transactions to the statement. Open this page using **Financial, Statement** and then navigate to the **Details** tab.

Description of Page

Statement is a concatenation of summary information about the statement. It is composed of the name of the statement person, the statement construct print description, the create date of the statement and the statement status.

Statement ID is this statement's unique system-generated identifier.

If a statement has a large number of service agreements that contributed financial transactions to the statement, you can use the **SA Filter** to limit the service agreements that appear in the grid. The following options are available:

- **All.** Use this option to view all service agreements that contributed financial transactions to the statement.
- **Person Name.** Use this option to only show service agreement linked to accounts whose main customer has a primary name that matches **Person Name**.
- **SA's at an Address.** Use this option to only show service agreements that are linked to service points associated with a given **Address, City** and/or **Postal** code. Note, you can specify any combination of these fields.
- **Statement Print Description.** Use this option to only show service agreements associated with a construct detail with a given **Statement Print Description**.

Don't forget to click the search button after changing the filter.

The grid contains service agreements that contributed financial transactions to the statement. If a service agreement's financial transactions appeared on multiple bills, a separate line is displayed for each bill. You can press the go to button adjacent to **Bill Due Date** to be transferred to [Financial Transactions On A Bill](#) where the individual financial transactions associated with the associated service agreement and bill can be viewed. The following information is displayed in the grid:

- **Statement Print Description** is the description of the construct detail associated with the service agreement.
- **Bill's Due Date** is the date of the bill on which the service agreement's financial transactions appear. Press the adjacent go to button to be transferred to [Financial Transactions On A Bill](#) where the individual financial transactions can be viewed.

- **Premise Information** describes the characteristic premise, if any, associated with the service agreement.
- **SA Information** describes the service agreement that contributed financial transactions to the statement.
- **Account Information** describes the account associated with the service agreement whose details appear on the statement.
- **Construct Detail ID** is the unique identifier of the construct detail associated with the service agreement.
- **Statement Detail ID** is the unique identifier of the statement detail on which the service agreement's financial transactions appear.

Sales & Marketing

The sales and marketing functionality satisfies many diverse requirements. For example, you can use this functionality to:

- Enroll new customers using a single transaction (i.e., you don't have to use the person, premise, service point, and start / stop transactions to enroll a new customer who resides at a new premise).
- Sell new products to existing customers.
- Update person, account and premise information using a single transaction.
- Market your services to prospects from a marketing list (and measure the success of your efforts). If the customer responds to your sales efforts, the system will automatically setup the customer, premise and related service agreements.
- Setup marketing surveys and record your customers' responses.
- Quickly create one-time charges.
- Setup [proposals](#) for prospective services (and then send a quotation to the customer for these services).
- And more...

The topics in this section provide more information about the sales and marketing functions.

Contents

[The Big Picture of Campaigns, Packages and Orders](#)
[The Big Picture Of Package Eligibility Rules](#)
[The Big Picture Of Campaign Eligibility Rules](#)
[Designing Campaigns and Packages](#)
[Other Useful Information](#)
[Maintaining Orders](#)
[Maintaining Campaigns](#)
[Maintaining Packages](#)

The Big Picture of Campaigns, Packages and Orders

Three objects are at the heart of the sales and marketing functionality: **Campaigns**, **Packages**, and **Orders**.

- A **package** defines a basket of goods and services that can be offered to a customer or a prospect.
- A **campaign** is a structured effort to offer a given set of packages to existing customers or new prospects.
- An **order** is created for each customer / prospect to which a campaign is targeted.

The topics in this section provide background information about these objects.

Contents

[An Overview of Campaigns, Packages and Orders](#)
[Supported Business Processes](#)
[Examples of Campaigns and Packages](#)

An Overview of Campaigns, Packages and Orders

An order is used to define:

- Demographic information about a customer / prospect.
- Geographic information about the service address.
- The customer's response to eligibility-oriented questions. For example, you can pose questions like: Who is your current energy service provider? / Would you like to pay automatically? / What is your date of birth?

After the above information is defined, the [order transaction](#) presents packages that may be offered to the customer. A package controls the various types of service agreements that will be created if the customer selects the package.

The customer must be eligible for a package. When you setup a [package](#), you define its [eligibility criteria](#). For example, you can setup a package that is only applicable to commercial customers in Toronto who pay automatically and who commit to a one-year service contract.

If the customer elects to take a package, the order transaction sets up / updates [the "V"](#) (along with all of the ancillary things that happen when service is initiated, e.g., field activities are created). Please note that in addition to setting up the "V", it's also possible to populate / update other information when a package is selected. For example, you could have the system setup the customer's automatic payment options.

An order can be completed without creating service agreements. It is possible to use the order transaction to simply create / update persons and accounts. Refer to [Marketing Surveys](#) and [Setting up a New Customer Prior To Using Start/Stop](#) for more information.

All orders must reference a campaign. An order's [campaign](#) defines:

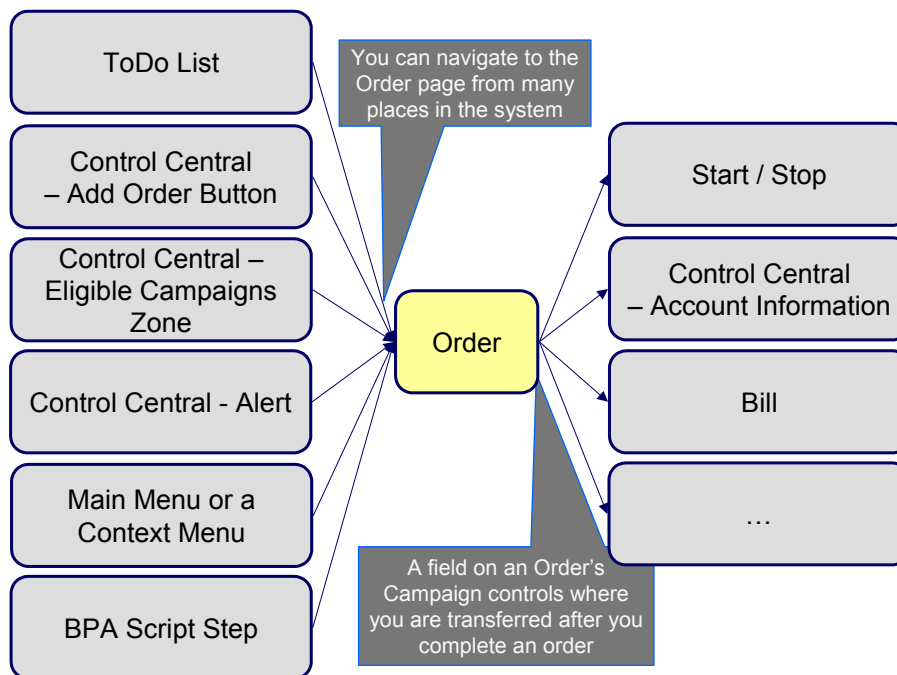
- How the campaign's orders use accounts and premises. For example, you can indicate that a given campaign is only targeted at existing premises (thus preventing the creation of a new premise when an order is completed).
- The type of information defaulted onto an order. For example, you can setup a campaign to default a given account management group on all orders linked to the campaign. This account management group will subsequently default onto the new account when the order is completed.
- The validation rules that control how its orders use accounts and premises. Specifically, on a campaign you define if an account / premise is required / optional / not allowed on its orders. In addition, if an account / premise is required or optional, you can control whether new accounts / premises can be created when an order is completed (the alternative is to force each order to use an existing account / premise). These controls prevent the unwanted proliferation of new accounts and premises for campaigns that are targeted at existing accounts and premises.

- The eligibility-oriented questions that are posed to the customer when an order is taken. For example, the questions indicated above – Who is your current energy service provider? / Would you like to pay automatically? / What is your date of birth? – are all defined on the order's campaign.
- The superset of packages that can be offered to a customer whose order references this campaign. An order's campaign defines the types of packages that may be selected.
- In addition to the above, campaigns also control high-level [eligibility rules](#) and [business process flows](#).

Bottom line. Campaigns allow you to define a group of **packages** that a customer is eligible to purchase. Every time a campaign is targeted at a customer, an **Order** is created. The information entered on the order qualifies the customer for one of the campaign's packages. If a customer elects to take a package, [the "V"](#) is setup / updated along with all of the ancillary things that happen when service is initiated, e.g., field activities are created, workflow processes are initiated, etc.

Supported Business Processes

Before providing examples of specific business processes, we'd like to highlight that you have control over some aspects of the user-interface flow (i.e., the screen navigation is not hard-coded). The following illustration highlights the potential user-interface flow:



Bottom line. You can navigate to the order transaction from many places in the system. After you complete an order, you are transferred to an appropriate transaction. The specific transaction is controlled by you when you setup the order's [campaign](#).

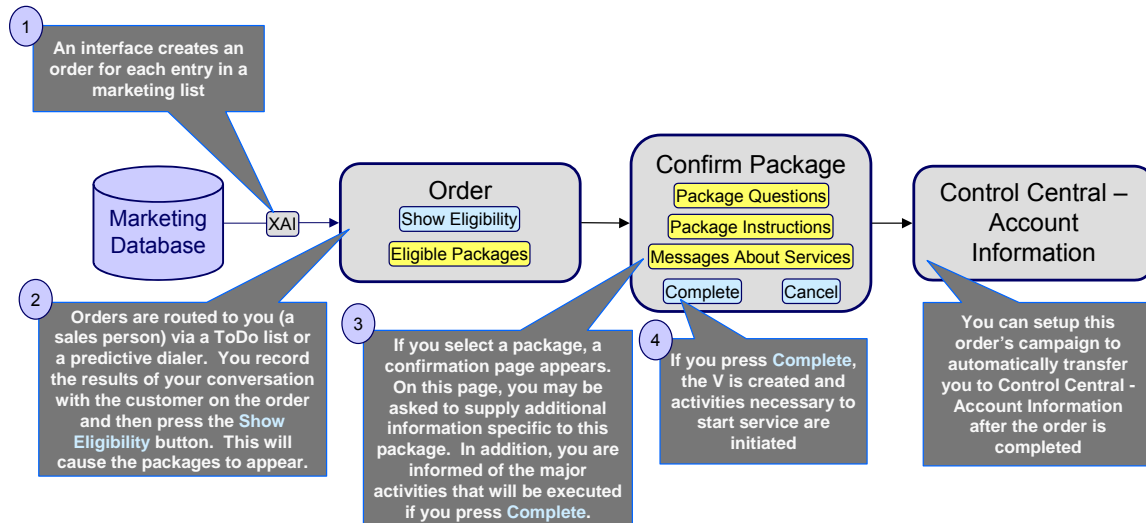
The topics that follow illustrate specific business process flows that are possible in your implementation of the sales and marketing functionality.

Contents

- Marketing to Prospects from a Marketing List
- Marketing to Pre-selected Customers
- Real-time Marketing of Additional Services to a Customer
- Real-time Marketing of Services to a Prospect
- Setting up a New Customer Prior To Using Start/Stop

Marketing to Prospects from a Marketing List

The following business process flow illustrates how the sales and marketing functionality would be used to market to prospects uploaded from a marketing database.



The following points describe the steps in this business process:

- An upload interface creates an [order](#) for each entry in a marketing database.

Note. The system is not supplied with a dedicated interface to upload a marketing list. We recommend that you use the [XAI utility](#) to implement such an upload.

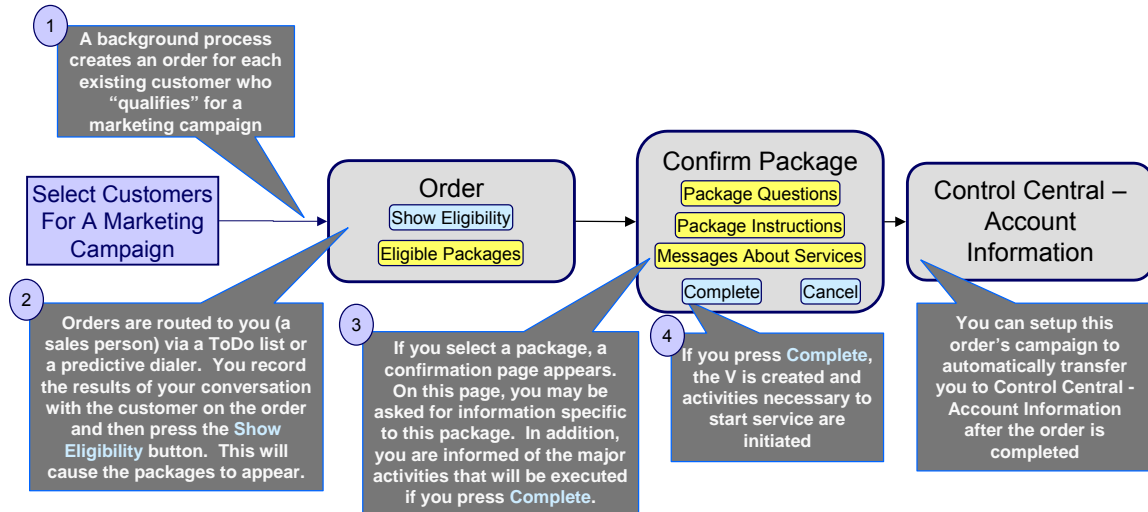
- Each [order](#) is routed to a salesperson via a [To Do entry](#) (or via a predictive dialer if your company supports this type of technology).
- The first step in the salesperson's interaction with the prospect will be to confirm the information uploaded from the marketing list (e.g., is your Bob Smith?, is your address 15 Main St?, etc.). Any changes are made to the information saved on the order.
- The salesperson's next step involves posing eligibility-oriented questions related to the campaign. Examples of questions include:
 - **What is your date of birth?** This type of question might be asked if you have packages that are only marketed to senior citizens. Potential answers would be a valid date that's in the past.
 - **Would you be interested in our full-service option (electricity, digital cable and gas)?** Potential options are: yes and no.

- **Who is your current electric service provider?** Potential answers are: Utility.Com, Reliant Resources, Other.
- **Would you like to pay automatically?** Potential answers are: Yes, No.
- **Would you like to pay using a levelized payment plan?** Potential answers are: Yes, No.
- Etc.
- The gathered information is saved on the order and then the system displays the packages that can be offered to the customer. The available packages may be restricted based on information on the order. For example, some packages are only applicable to senior citizens, while others are only applicable to customers whose current service provider is Utility.com.
- If the customer is interested in a package, the salesperson selects it. Prior to the order being completed, the [package confirmation](#) page is shown. On this page are additional instructions about the package and a description of how the system will be updated if the package is selected (e.g., the types of service agreements will be displayed). In addition, the package may require additional information before it can be chosen. For example, the package may only be selectable if the customer agrees to pay automatically. In this situation, the package confirmation page will require the salesperson to enter the customer's bank and bank account.
- After entering all package-specific information, the order is completed. When the order is completed, [the "V"](#) is setup and all work necessary to satisfy the order is initiated (e.g., field activities are created, workflow processes are initiated, etc.).
- And finally, the user is returned to [Control Central - Account Info](#) where an overview of the customer is displayed. From this page, the user can drill down to any service agreement (or to start/stop) to perform any fine-tuning.

Please be aware that the page to which you are transferred after order completion is controlled by a field on the order's [campaign](#). In our example, it made sense to transfer the user to Control Central as it provides a nice confirmation to the user (it shows a great deal of information about an account). However, you could setup the campaign to take the user to virtually any page in the system. You'll see good examples later in this discussion.

Marketing to Pre-selected Customers

The following business process flow illustrates how the sales and marketing functionality would be used to market additional services to existing customers selected by a background process.



The following points describe the steps in this business process:

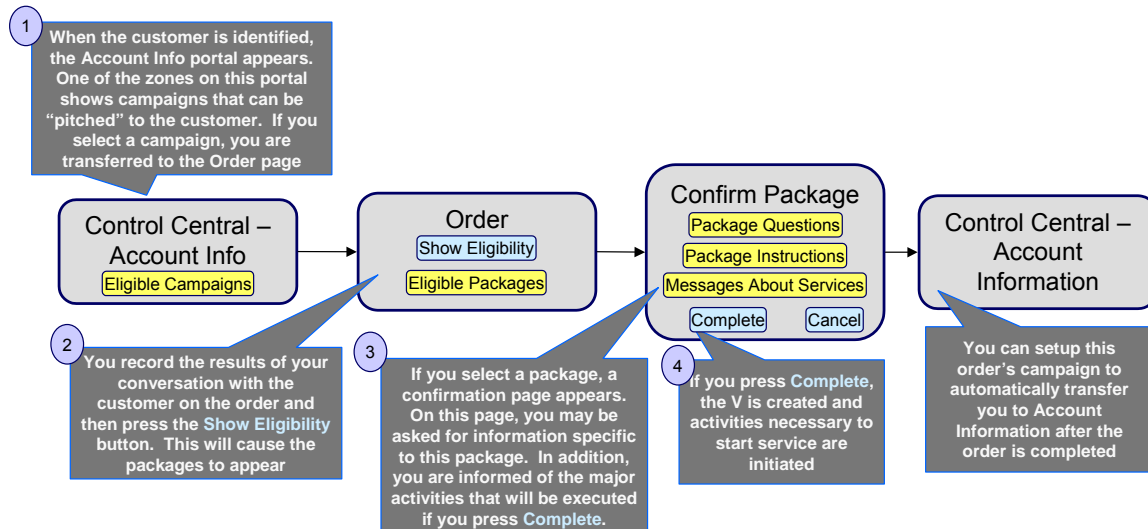
- A background process creates an [order](#) for each customer to be targeted under a marketing campaign.

Note. The base package does not have an example of such a background process. Please speak to your support team if you require assistance in developing this type of process.

- Sales people would then process the order as described [above](#).

Real-time Marketing of Additional Services to a Customer

The following business process flow illustrates how the sales and marketing functionality would be used to market additional services to existing customers when they call in.



The following points describe the steps in this business process:

- When you find the customer's record on [Control Central - Main](#), you are automatically transferred to the [Account Information](#) portal. One of the zones on this page contains the campaigns that can be offered to the customer.

Note. The “potential campaigns” zone only appears if the CSR has modified their preferences to display this zone.

- If you select a campaign from this zone, you are transferred to the [Order](#) transaction.

Note. An order's campaign controls whether the [Main](#) or [Questions & Misc Fields](#) tab is initially displayed when the order transaction is invoked from this zone. For example, you might want to have the **Main** tab displayed if you want the order-taker to confirm the customer's demographic information before posing questions. Alternatively, you may want to skip this step and immediately present the order-taker with the campaign's questions. Keep in mind that the user can always display the other tab regardless of what is initially displayed.

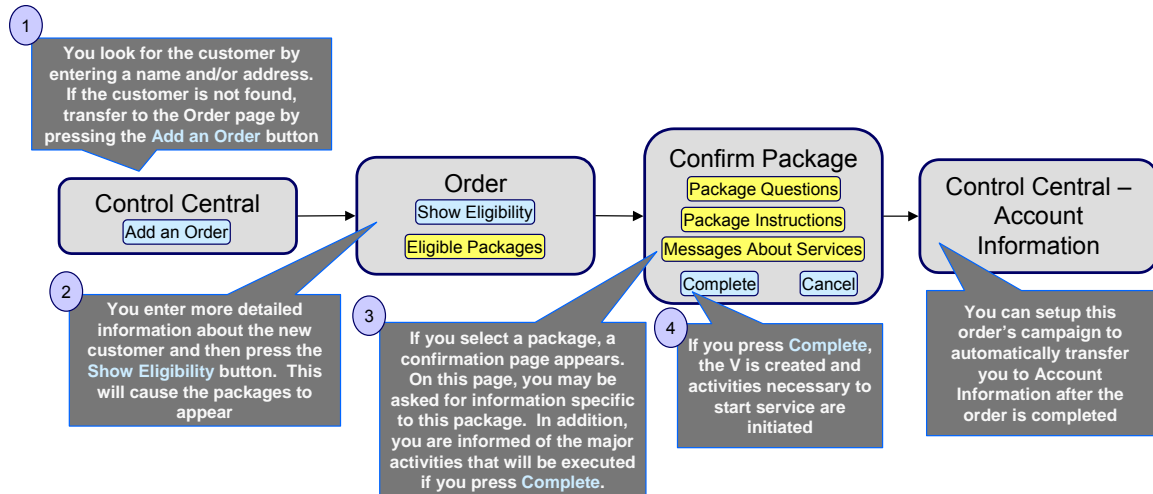
- You then process the order as described [above](#).

An easy way to create one-time charges. The above business process illustrates how the selection of a package will create one or more new service agreements for a customer. If your organization levies one-time charges (e.g., tree trimming charges, damage assessments, etc.), you can also use the order transaction to create both a “one-time charge” service agreement and create the related billable charge when the order is completed. To do this, setup a campaign with a separate package for each possible one-time charge. If an order is created for this type of campaign and the relevant package is selected, the system will setup the new billable charge service agreement and link the billable charge to it. Refer to [An Easier Way To Create One Time Charges](#) for an example of such a campaign.

Real-time Marketing of Services to a Prospect

The following business process flow illustrates how the sales and marketing functionality would be used to market services when a new customer calls.

Warning! We do not recommend using this business process if your organization's service territory is predefined (i.e., if you've set up the premises and service points for your entire service territory). Why? Because it can result in the creation of new premises and / or service points and you probably don't want this to happen if you've already set everything up. While it is possible to disable the creation of new service points and premises by developing new plug-ins, we'd recommend using the business process described under [How To Add A New Customer From Control Central](#) if a new customer calls in and you have predefined premises and service points.



The following points describe the steps in this business process:

- You look for the customer using [Control Central - Main](#). Because the customer is new, you push the Add Order button on Control Central to transfer to the [Order](#) transaction.
- You then process the order as described above.

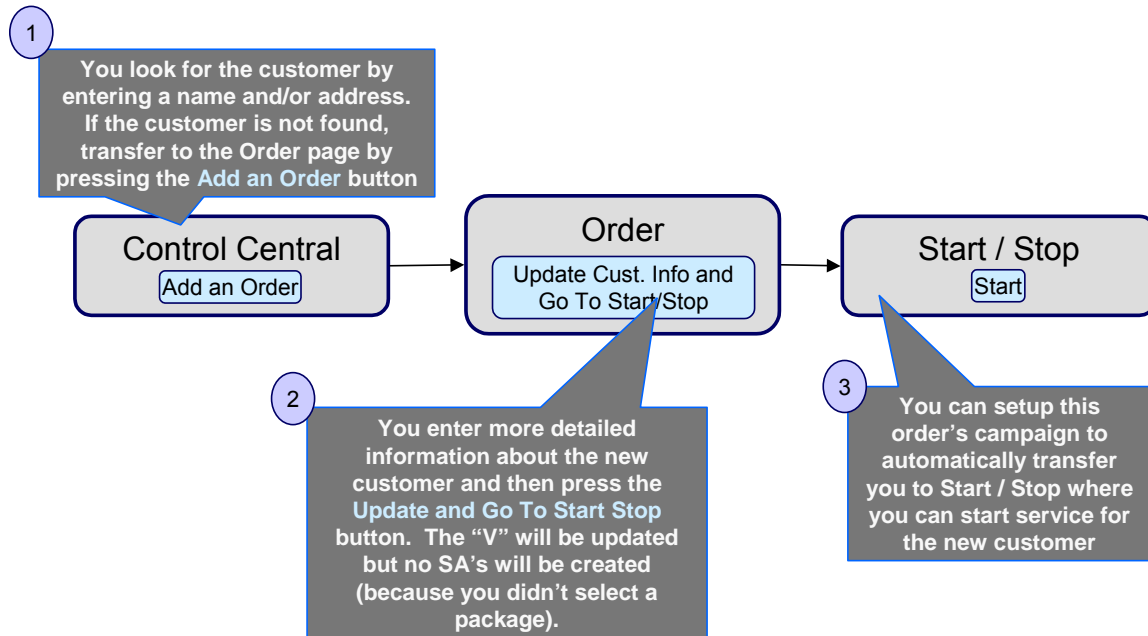
In order to take advantage of the above business process, it's important that you understand the following points:

- When the order page is opened for a new customer, the order's campaign is defaulted from the [installation record](#). The default campaign on the installation record is "generic" because we don't know anything about the customer (they are new) and therefore we cannot default a campaign targeted to their customer class or geography.
- This default campaign on the installation record must be constructed using one of the following approaches:
 - The default campaign has a package for every potential type of customer who can call up. While this is possible, it's probably not realistic as the data setup burden required to define every type of package could be onerous. However, if you have a limited number of service agreements, this approach is feasible.
 - The default campaign could just be a "dummy" campaign that has no packages. The "real" campaigns will only be shown after basic information about the customer has been recorded on the order and the "show eligibility" button is pressed. These campaigns appear in the order's [eligibility tree](#). The selection of a campaign in this tree will cause the order's campaign to be switched. Note, because the "dummy" campaign has no packages, no packages will appear in the order's eligibility tree when the dummy campaign is referenced on an order.
- The order is then processed as described [above](#).

Setting up a New Customer Prior To Using Start/Stop

The following business process flow illustrates how the sales and marketing functionality would be used to create a new customer prior to using [Start / Stop](#) to start service at an existing premise.

Note. The following business process is similar to that described under [How To Add A New Customer From Control Central](#). The only difference is that the [Order](#) transaction is used to create a new person and account rather than the [Person](#) transaction. The order transaction may be a better choice because it lets you define many attributes on person, account and premise that are not accessible on the person page. For example, the order transaction lets you define the customer's automatic payment option, bill after date, override due date, etc. In addition, on an order's campaign, you can define default values of many field values (both real fields and characteristics). These default values are presented on the order transaction and may be overridden by the user at will.



The following points describe the steps in this business process:

- You look for the customer using [Control Central - Main](#). Because the customer is new, you push the "Add Order" button on Control Central to transfer to the [Order](#) transaction.
- You record additional demographic and geographic information about the new customer and confirm all values defaulted from the order's campaign. After all required information is entered, you press the Update Cust. Info and Go To Start / Stop button to create the new person and account and transfer to [Start / Stop](#) to identify the new services desired by the customer.

The button's existence and label are dynamic. In the above example, we indicated that there is a button on the order page called **Update Cust. Info and Go To Start / Stop**. This button's presence on the page and its label are controlled by fields on the [campaign](#). When pressed, the "V" is updated with information on the order and the user is transferred to the "post completion" page (which is also defined on the campaign). You would only enable this button on campaigns whose orders can be completed without selecting a package. In the above example, the campaign doesn't have any packages (because service is intended to be started using Start / Stop). In [Examples of Campaigns and Packages](#) we show other examples of how these fields would be setup for different types of campaigns.

In order to take advantage of the above business process, it's important that you understand the following points:

- When the order page is opened as a result of pressing the Add Order button on [Control Central - Main](#), the order's campaign is defaulted from the [installation record](#). The default campaign on the installation record is "generic" because nothing is known about the new customer and therefore it is impossible to default a campaign targeted to their customer class or geography.
- The campaign on the installation record should be setup to contain default information for the "average" customer. For example, if you typically deal with residential customers, the information on the default campaign should be residential-oriented.
- When a user takes an order for a new customer, they'll change the default information when appropriate. For example, if an industrial customer were being added, they'd change the customer class to "industrial" and then ask to see the campaigns such a customer is eligible to choose (these alternate campaigns are displayed in the order's [eligibility tree](#)). For example, if you've setup campaigns with [eligibility rules](#) that are targeted at industrial customers and the customer class on the order has been changed to "industrial", the user will see these campaigns in the eligibility tree. The user can then change the campaign's order by simply clicking on the desired campaign.

Note. The above example illustrates how an order can be completed without causing the creation of service agreements. Refer to [Marketing Surveys](#) and [Prelude to Start Stop](#) for examples of such campaigns.

Examples of Campaigns and Packages

The topics that follow provide examples of [campaigns](#) and [packages](#) that support several different scenarios. Your organization's campaigns and packages may differ markedly from these examples. Factors that will influence your sales and marketing objects include:

- Whether the type of goods and services marketed to your customers / prospects differs depending on some trait of the customer or the region in which they are located. For example,
 - Your company may have very straightforward packages that only differ based on whether the customer is industrial, residential or commercial. If this describes your company, the setup process will be simple.
 - Alternatively, your company might offer different packages to residential customers based on whether they pay automatically, whether the customer is a senior citizen, whether they purchase multiple services from you, etc. If this describes your company, then you'll have to define the various eligibility criteria and the respective packages applicable to customers who satisfy each criterion.
- Whether premises exist prior to order taking. For example, a company that distributes a commodity has a well-defined geographic area and therefore new premises typically don't arise when orders are taken.
- Whether your company offers many different types of goods and services.
- Whether your rates have charges / discounts that are only applied to some customers. For example, you might have discounts that are applied if the customer is a senior citizen or an employee.

- Which of the [Supported Business Processes](#) you implement.

Use the information that follows to form an intuitive understanding of campaigns and packages. After attaining this understanding, you'll be ready to design your own campaigns and packages.

Contents

- [Campaigns Without Packages](#)
- [Campaigns With Simple Packages](#)
- [Campaigns With Complex Packages](#)

Campaigns Without Packages

An order that references a campaign without packages will never create service agreements. The following topics describe scenarios when such a campaign would be used.

Contents

- [Marketing Surveys](#)
- [Prelude to Start/Stop](#)

Marketing Surveys

Some organizations conduct “marketing surveys” to determine general trends amongst their customer base. The following diagram illustrates how such a campaign might look:

Campaign – Residential Survey

Default Info For New Customers / Premises

Not applicable

Questions To Be Posed On Orders

Question	Possible Answers
Electric or gas water heater?	Elec, Gas
How many low-flow toilets?	1 through 9
How many people are in your household?	1 through >12
How would you rate our service?	1 (bad), 2 (OK), 3 (great)
How many outages have you experienced in the last year?	0 through >12

Misc Fields To Be Captured

Not applicable

Campaign Eligibility Rules

Algorithm

Customer class is residential

Potential Packages

Not applicable

Campaign Behavior Controls

Enable button that allows order completion without package selection	Yes
Button Label	Complete Survey
Post completion transaction	Account Info

Note the following about this campaign:

- If the campaign is used to establish new customers, you can define information to default onto orders created for new customers. Because marketing surveys are targeted at existing customers, you don't need new customer default information on the campaign.
- You can define questions to be posed to customers when an order is taken when you setup a campaign. For a "marketing survey" campaign, you'd define each of the survey questions and the permissible answers. When a customer is surveyed, an order is created and their answers are saved on the order.

- On a campaign, you can define additional fields to be updated on persons, accounts and premises when orders associated with the campaign are completed. For example, you could indicate that the account's override due date and automatic payment options should be confirmed when an order is taken. Because marketing surveys are typically only asking questions, you wouldn't need to define additional customer information fields to be confirmed. However, you can if you want.
- All campaigns need at least one eligibility rule. These rules control whether the campaign appears in the "eligible campaigns" [content zone](#) when a customer is selected on Control Central. These rules also control whether the campaign appears on an order's [eligibility tree](#) (campaigns that appear in this tree are alternate campaigns that may be used on the order). Our example has assumed that this is a [residential](#) customer survey and therefore it only needs a single eligibility rule. Refer to [The Big Picture Of Campaign Eligibility Rules](#) for more information.
- Because this campaign is used purely to manage a marketing survey, it doesn't need any packages. Packages are only necessary if service agreements are created when an order is completed.
- Because this campaign doesn't have packages, we have indicated that the button that allows orders to be completed without selecting a package is enabled. If you don't enable this button, a user won't be able to complete a marketing survey (and it is only during order completion when the "V" is updated with the information on the order). Notice that we have assigned the button a label of Complete Survey.
- This campaign should be setup to transfer the user to [Control Central - Account Info](#) after the survey is completed.

Prelude to Start/Stop

Consider the example illustrated above under [Setting up a New Customer Prior To Start / Stop](#). In this example, the [order](#) transaction is used to create a new person and account; no service agreements are created when the order is completed. Rather, after the order is completed, the user is automatically transferred to [Start / Stop](#) where they can create service agreements. The following diagram illustrates how such a campaign might look:

Campaign – Add New Customer

Default Info For New Customers / Premises

Field	Default Value
Customer Class	R
Account Management Group	RESID

Questions To Be Posed On Orders

Not applicable

Misc Fields To Be Captured

Field	Required / Optional	Default Value
Auto-pay source	Optional	‘ ‘
Auto-pay account	Optional	‘ ‘
Date of birth	Optional	‘ ‘

Campaign Eligibility Rules

Algorithm
This campaign is only used to create new customers

Potential Packages

Not applicable

Campaign Behavior Controls

Enable button that allows order completion without package selection	Yes
Button Label	Update the V / Go To Start
Post completion transaction	Start/Stop

Note the following about this campaign:

- Because this campaign is used to establish new customers, you should define new customer default information. The order taker can override these default values.
- If we assume you don't ask marketing / eligibility-oriented questions when you take-on a new customer then this campaign would not require any questions.

- On a campaign, you can define additional fields to be updated on the related person, account and premise when orders associated with the campaign are completed. For example, you could indicate that the account's override due date and automatic payment options should be confirmed when an order is taken. In our example, we are assuming that we want to define the customer's automatic payment options and his/her date of birth.
- As described under [Marketing Surveys](#), a campaign's eligibility rule(s) control whether the campaign appears in the "eligible campaigns" [content zone](#) and on an order's [eligibility tree](#). This campaign is odd in that we don't want it to ever appear as an "eligible campaign" because it's only used for NEW customers. In order to make this work you must:
 - Define an eligibility algorithm on this campaign that prevents it from appearing as an "eligible campaigns". Refer to [A Campaign That Is Never Eligible](#) for more information.
 - Update the [installation record](#) to indicate that this campaign is the one that should be used on orders created when a user presses the "Add an Order" button on [Control Central - Main](#). Refer to [Setting up a New Customer Prior To Starting Service](#) for more information about the flow of this business process.
- Because this campaign is used purely to add accounts and persons, it doesn't need any packages. Packages are only necessary if service agreements are created when an order is completed.
- Because this campaign doesn't have packages, we have indicated that the button that allows an order to be completed without selecting a package is enabled. If you don't enable this button, a user won't be able to complete the order (and it is only during order completion when the "V" is updated with the information on the order). Notice that we have assigned the button a label of Update The V and Go To Start.
- This campaign should be setup to transfer the user to [Start / Stop](#) after the order is completed.

Campaigns With Simple Packages

As explained under [Campaigns Without Packages](#), there are several situations that might necessitate a campaign without packages. In this section, we describe simple scenarios of campaigns with packages.

An Easier Way To Create One Time Charges

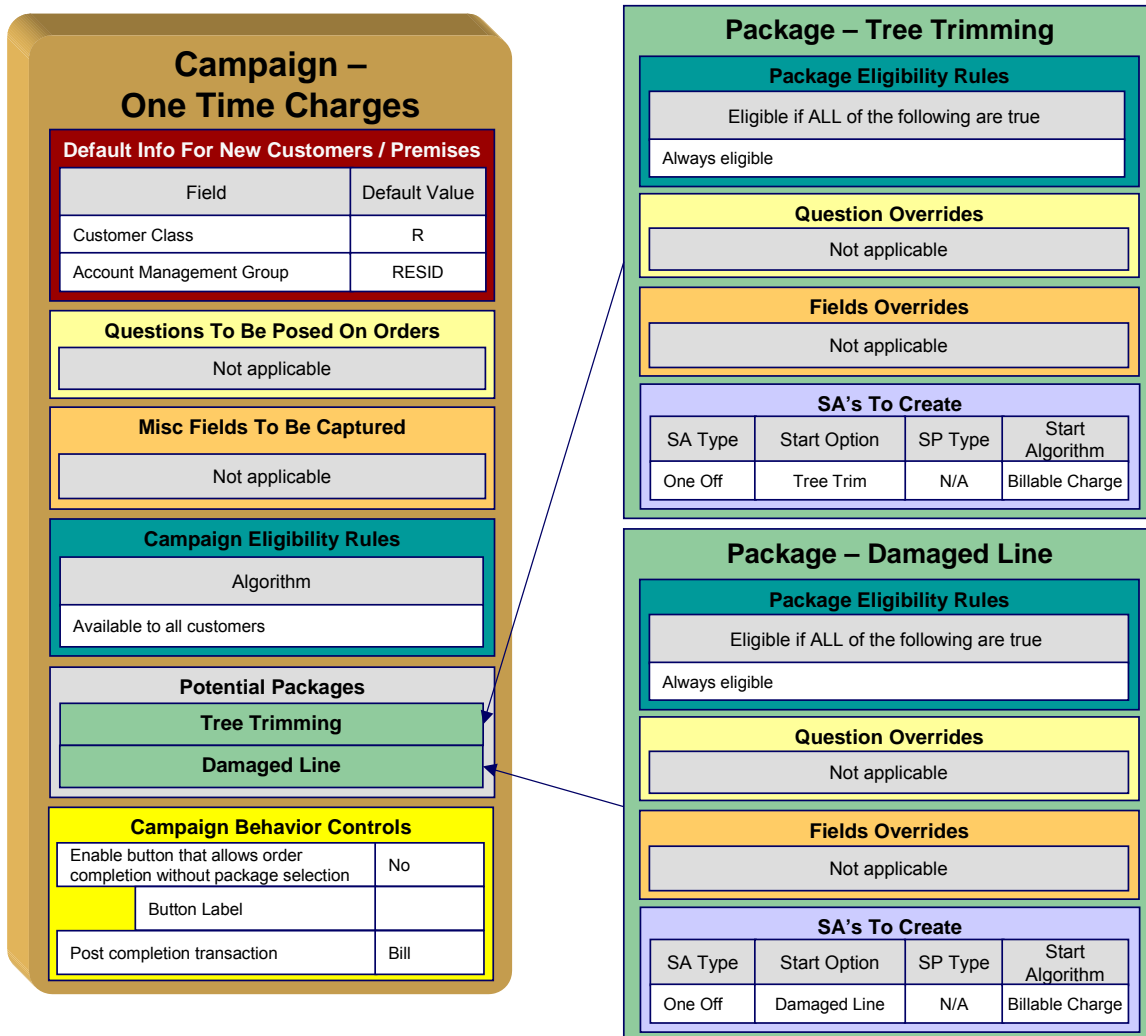
We use the term "one-time charge" to describe ad hoc charges, e.g., tree trimming charges, damage assessments, etc. The following points describe how one-time charges are levied if you don't use the order transaction:

- Determine if the customer to be invoiced already has an appropriate "one-time charge" service agreement. If one doesn't exist, you must create one using [Start/Stop Service](#).
- Next, a billable charge is added to the "one-time charge" service agreement (using [Billable Charge Maintenance](#)). The billable charge contains the invoice lines and amounts. The system will produce the bill for this charge when the account's bill cycle next executes.

Rather than going through the above multi-step process, you can also use the order transaction to create both a "one-time charge" service agreement AND create the billable charge. In addition, if the customer receiving the charge is new, the order transaction will also create a new person and account at the same time.

To take advantage of this facility, you must set up a campaign that has a separate package for each type of one-time charge as illustrated below. When an order is taken for such a campaign, the user simply selects the package that corresponds with the desired one-time charge. The system will then create the necessary customer and financial information when the package is confirmed.

Billing of one-time charges. The order transaction will not cause a bill to be generated for the customer. Rather, the next time the account is billed, the system will see that there's an unbilled billable charge and create a bill segment for it. This bill segment will appear on the customer's regular bill along with their other charges. If you'd prefer to create the bill for the one-time charges immediately, you could setup the campaign to automatically transfer the user to the bill transaction where they can [create an ad hoc bill](#). Note, if you do this, you'll need to create a new plug-in algorithm on the billable charge SA type to automatically activate the service agreement when it is created (otherwise the user would have to manually activate the billable charge service agreement before the bill can be generated).



Note the following about this campaign:

- One-time charges can be levied against both new and existing customers. This means that orders associated with this campaign may cause new customers to be created. Therefore, we have specified information to default onto orders for new customers on the campaign. Note that we defined “residential-oriented” default values as the majority of one-time charges are levied against residential customers.
- If we assume you don’t ask marketing / eligibility-oriented questions when you levy a one-time charge then this campaign would not require any questions.
- We have assumed that you are not capturing miscellaneous fields when you levy a one-time charge.
- As described under [Marketing Surveys](#), all campaigns need at least one eligibility rule. We have assumed that this campaign’s packages can be selected by all types of customers and have therefore specified an eligibility algorithm that will be true for all customers.
- Because this campaign has packages, we have indicated that the button that allows an order to be completed without selecting a package is disabled. This means a user must choose one of the packages to complete the order.
- You might want to setup this campaign to automatically transfer the user to the bill transaction after the order is completed. See the note above the illustration for more information.
- We have assumed that this campaign can be used to levy two kinds of one-time charges – a tree trimming charge and a damaged line charge – and therefore this campaign requires two packages. Note the following about each package:
 - A package’s eligibility rules control whether the package can be selected when an order is taken. With more sophisticated campaigns, you may have packages that are only applicable to certain types of customers. In our example, we’ve assumed that these types of one-time charges can be levied against all customers and have therefore used “universal” eligibility criteria (i.e., there are no eligibility restrictions). Refer to [A Package That Is Always Eligible](#) for more information.
 - On a package, it’s possible to define additional questions to be posed if a package is selected. In our example, there are no additional questions to be posed for the one-time charge packages.
 - On a package, it’s possible to define additional fields to be captured if a package is selected. In our example, there are no additional fields to be captured.
 - On a package, you define the types of service agreements to be created if a package is selected. In our example, we only need a single service agreement created / updated if the package is selected and therefore there’s just one entry in **SA’s To Create**. Please note the following about the information used to create this service agreement:
 - The **SA Type** is one that’s used for a [billable charge service agreement](#).
 - The SA Type’s **Start Option** is one that will automatically create a billable charge when a service agreement is created (if a [start option](#) references a [billable charge template](#), a billable charge will be created when this start option is used). In our example, we have assumed a different start option has been setup for each type of one-time charge.
 - Billable charge service agreements do not require service points and therefore **SP Type** is not applicable.
 - We’ve indicated a **Start Algorithm** of [Billable Charge](#). Refer to [STRM-AS](#) for an example of such an algorithm (type).

Campaigns With Complex Packages

The [Campaigns With Simple Packages](#) illustrated campaigns with very simple packages. In this section, we describe a campaign with more complex packages.

Contents

[Packages Limited By Answers And Field Values](#)

[Packages Limited By Current State Of Service](#)

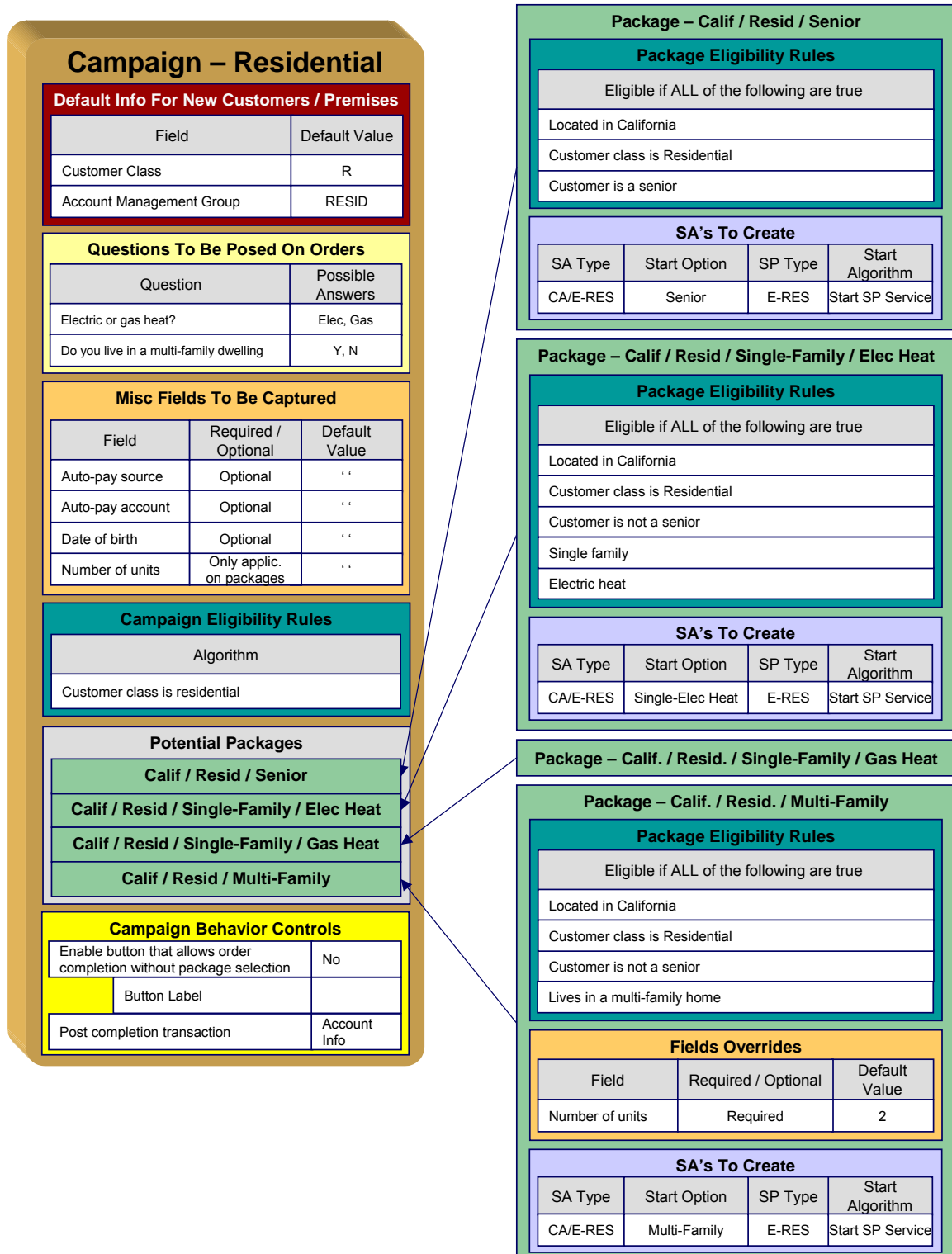
Packages Limited By Answers And Field Values

Assume your organization has a campaign with several packages. Each package has the following restrictions in respect of its use:

- “Package 1” can be offered to residential customers in California who are senior citizens.
- “Package 2” can be offered to residential customers in California who live in single-family homes who are not senior citizens and who heat their homes with electricity.
- “Package 3” can be offered to residential customers in California who live in single-family homes who are not senior citizens and who heat their homes with gas.
- “Package 4” can be offered to residential customers in California who live in multi-family homes who are not senior citizens.

Also assume that all customers are encouraged to pay automatically.

The following diagram illustrates how such a campaign might look:



The topics that follow describe the campaign and packages used to support this scenario. We'll start by explaining each package. Then we'll describe how the campaign will be setup. This seems backwards, but it mirrors how you should design this type of campaign:

Contents

Senior Citizen Package
Single-Family Electric Package
Single-Family Gas Package
Multi-Family Package
The Campaign

Senior Citizen Package

The package that is offered to residential customers in California who are senior citizens has the following traits:

- It has eligibility rules that restrict its use to customers who meet the following criteria:
 - Customer class = Residential
 - Division = California
 - Birth date equates to that of a senior citizen
- We only need a single service agreement created if the package is selected and therefore there's just one entry in SA's To Create. Please note the following about the information used to create this service agreement:
 - The **SA Type** is one that's used for an electric residential service agreement (hence the **E-RES** SA type).
 - The SA Type's **Start Option** is one that will automatically setup the service agreement with the appropriate rate and contract options for a senior citizen.
 - The service agreement needs to be linked to an electric residential service point (hence the **E-RES** SP type).
 - We've indicated a Start Algorithm of **Start SP Service**. Refer to [STRM-VT](#) for an example of such an algorithm (type).
- We have not shown additional questions or miscellaneous fields in this package. This is because this package doesn't have additional questions or additional fields.

Single-Family Electric Package

The package that is offered to residential customers in California who live in single-family homes who are not senior citizens and who heat their homes with electricity has the following traits:

- It has eligibility rules that restrict its use to customers who meet the following criteria:
 - Customer class = Residential
 - Division = California
 - Birth date does not equate to that of a senior citizen
 - The premise is a single-family home
 - The premise is heated with electricity
- We only need a single service agreement created if the package is selected and therefore there's just one entry in SA's To Create. Please note the following about the information used to create this service agreement:
 - The **SA Type** is one that's used for an electric residential service agreement (hence the **E-RES** SA type).

- The SA Type's **Start Option** is one that will automatically setup the service agreement with the appropriate rate and contract options for this type of customer.
- The service agreement needs to be linked to an electric residential service point (hence the **E-RES** SP type).
- We've indicated a Start Algorithm of **Start SP Service**. Refer to [STRM-VT](#) for an example of such an algorithm (type).
- We have not shown additional questions or miscellaneous fields in this package. This is because this package doesn't have additional questions or additional fields.

Single-Family Gas Package

The package that is offered to residential customers in California who live in single-family homes who are not senior citizens and who heat their homes with gas has the following traits:

- It has eligibility rules that restrict its use to customers who meet the following criteria:
 - Customer class = Residential
 - Division = California
 - Birth date does not equate to that of a senior citizen
 - The premise is a single-family home
 - The premise is heated with gas
- We only need a single service agreement created if the package is selected and therefore there's just one entry in SA's To Create. Please note the following about the information used to create this service agreement:
 - The **SA Type** is one that's used for a gas residential service agreement (hence the **G-RES** SA type).
 - The SA Type's **Start Option** is one that will automatically setup the service agreement with the appropriate rate and contract options for this type of customer.
 - The service agreement needs to be linked to a gas residential service point (hence the **G-RES** SP type).
 - We've indicated a Start Algorithm of **Start SP Service**. Refer to [STRM-VT](#) for an example of such an algorithm (type).
- We have not shown additional questions or miscellaneous fields in this package. This is because this package doesn't have additional questions or additional fields.

Multi-Family Package

The package that is offered to residential customers in California who live in multi-family homes who are not senior citizens has the following traits:

- It has eligibility rules that restrict its use to customers who meet the following criteria:
 - Customer class = Residential
 - Division = California
 - Birth date does not equates to that of a senior citizen
 - The premise is a multi-family home

- This package requires additional information that was not requested at the campaign-level. We're going to assume that the multi-family rate needs to know the number of units in the building. This means that this package needs an additional field (number of units).
- We only need a single service agreement created if the package is selected and therefore there's just one entry in SA's To Create. Please note the following about the information used to create this service agreement:
 - The **SA Type** is one that's used for an electric residential service agreement (hence the **E-RES** SA type).
 - The SA Type's **Start Option** is one that will automatically setup the service agreement with the appropriate rate and contract options for this type of customer.
 - The service agreement needs to be linked to an electric residential service point (hence the **E-RES** SP type).
 - We've indicated a Start Algorithm of **Start SP Service**. Refer to [STRM-VT](#) for an example of such an algorithm (type).
- We have not shown additional questions in this package. This is because this package doesn't have additional questions.

The Campaign

Note the following about this campaign:

- Orders can be created for both new and existing customers. Because new customers are possible, we have specified information to default onto orders for new customers. Note that we defined "residential-oriented" default values (but remember, these values can be overridden on the order).
- The questions on the campaign are those whose answers play a part in each package's eligibility. You could also ask additional [marketing survey-oriented](#) questions that have nothing to do with package eligibility.
- The miscellaneous fields highlight an interesting situation. Notice that the date-of-birth and "automatic payment" fields are optional. However the field that holds the number of dwelling units is "only applicable on packages". When a field has this designation, the user will only be prompted to supply this field's value if a package is chosen where this field is applicable. In our example, only the [Multi Family Package](#) uses this field. This package will be setup to indicate that this field is required whereas the other packages will be setup to indicate that this field is not used.

Bottom line. The "miscellaneous fields" on a campaign must also include package-specific fields.

- All campaigns need at least one "highlight rule". These rules control whether the campaign appears in the "eligible campaigns" [content zone](#) when a customer is selected on Control Central. Also note that "highlight rules" also control whether the campaign appears on an order's eligibility tree (campaigns that appear in this tree are alternate campaigns that may be used on the order).

We're going to assume that this is a [residential](#) campaign and therefore you would specify a highlight rule that returns true if the customer's customer class is residential.

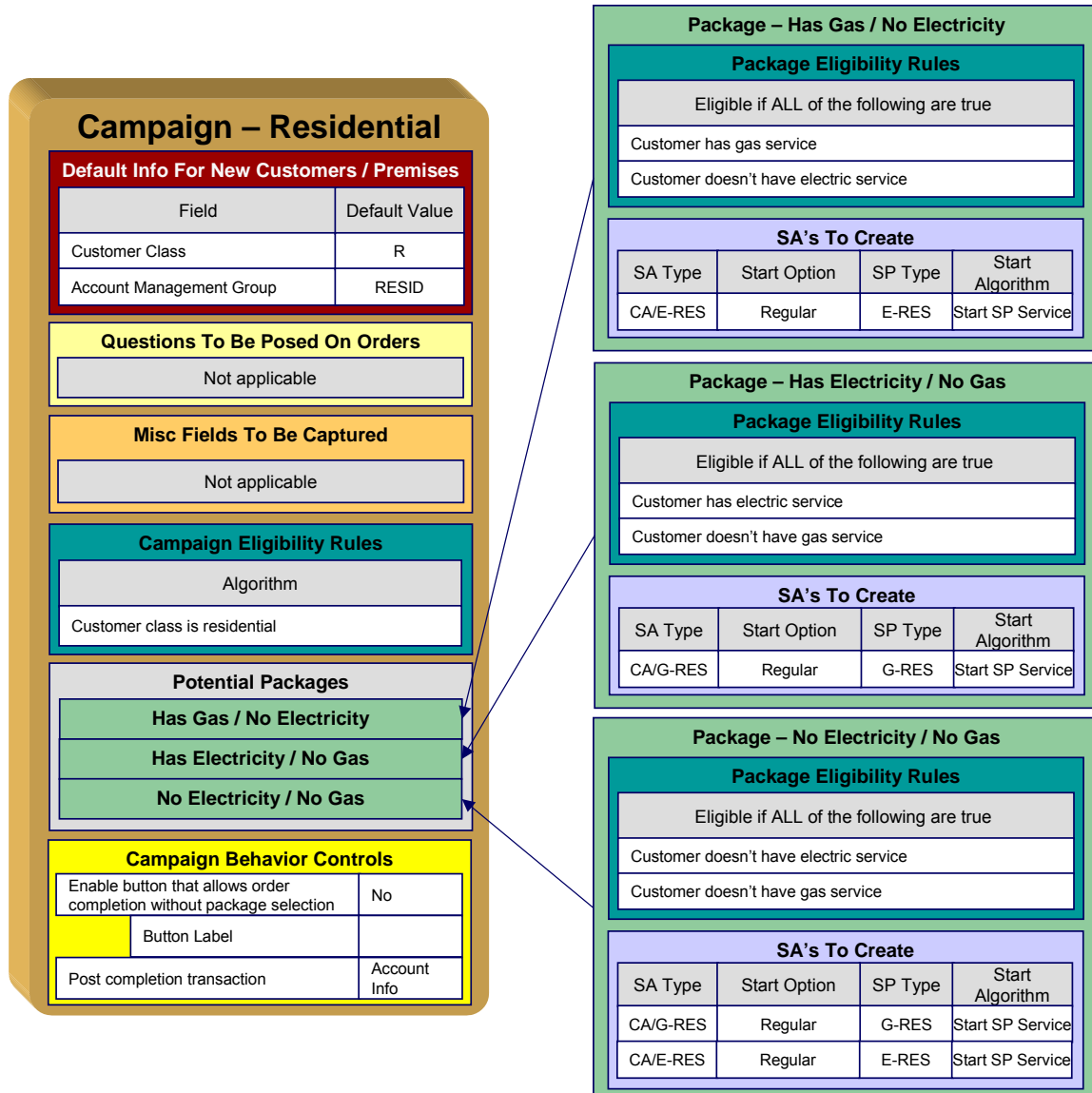
- Because this campaign has packages, we have indicated that the button that allows an order to be completed without selecting a package is disabled. This means a user must choose one of the packages to complete the order.
- This campaign is setup so that the user will be transferred to [Control Central - Account Info](#) after the order is completed. This is a subjective decision as you could transfer the user to a variety of transactions. It's just that the Account Info page provides a nice confirmation of how the customer looks after the order has updated / created new information.

Packages Limited By Current State Of Service

Assume your organization has a campaign with several packages. Each package has the following restrictions in respect of its use:

- "Package 1" can be offered to residential customers in California who have gas service, but don't have electric service.
- "Package 2" can be offered to residential customers in California who have electric service, but don't have gas service.
- "Package 3" can be offered to residential customers in California who have neither electric nor gas service.

The following diagram illustrates how such a campaign might look:



The topics that follow describe the campaign and packages used to support this scenario. We'll start by explaining each package. Then we'll describe how the campaign will be setup. This seems backwards, but it mirrors how you should design this type of campaign:

Contents

- Has Gas / No Electricity Package
- Has Electricity / No Gas Package
- No Electricity / No Gas Package

Has Gas / No Electricity Package

The package that is offered to customers that currently have gas service, but don't have electricity service has the following traits:

- It has eligibility rules that restrict its use to customers who meet the following criteria:
 - Customer has gas service

- Customer doesn't have electricity service

Refer to [A Package With Service Type Comparisons](#) for more information.

- We only need a single service agreement created if the package is selected and therefore there's just one entry in SA's To Create. Please note the following about the information used to create this service agreement:
 - The **SA Type** is one that's used for an electric residential service agreement (hence the **E-RES** SA type).
 - The SA Type's **Start Option** is one that will setup the service agreement with the appropriate rate and contract options.
 - The service agreement needs to be linked to an electric residential service point (hence the **E-RES** SP type).
 - We've indicated a Start Algorithm of **Start SP Service**. Refer to [STRM-VT](#) for an example of such an algorithm (type).
- We have not shown additional questions or miscellaneous fields in this package. This is because this package doesn't have additional questions or additional fields.

Has Electricity / No Gas Package

The package that is offered to customers that currently have electric service, but don't have gas service has the following traits:

- It has eligibility rules that restrict its use to customers who meet the following criteria:
 - Customer has electric service
 - Customer doesn't have gas service
- We only need a single service agreement created if the package is selected and therefore there's just one entry in SA's To Create. Please note the following about the information used to create this service agreement:
 - The **SA Type** is one that's used for a gas residential service agreement (hence the **G-RES** SA type).
 - The SA Type's **Start Option** is one that will setup the service agreement with the appropriate rate and contract options.
 - The service agreement needs to be linked to a gas residential service point (hence the **G-RES** SP type).
 - We've indicated a Start Algorithm of **Start SP Service**. Refer to [STRM-VT](#) for an example of such an algorithm (type).
- We have not shown additional questions or miscellaneous fields in this package. This is because this package doesn't have additional questions or additional fields.

No Electricity / No Gas Package

The package that is offered to customers that don't currently have electricity or gas service has the following traits:

- It has eligibility rules that restrict its use to customers who meet the following criteria:
 - Customer doesn't have electric service

- Customer doesn't have gas service
- We need two service agreements created (one for electricity, one for gas).
- We have not shown additional questions or miscellaneous fields in this package. This is because this package doesn't have additional questions or additional fields.

The Big Picture Of Package Eligibility Rules

The packages described under [Examples of Campaigns and Packages](#) have a variety of eligibility rules. Designing these rules can be easy or time-consuming; it all depends on the complexity of your business rules. We'll walk you through several examples to help you form an intuitive understanding of how to setup eligibility rules. Once you've acquired this intuition you'll be ready to design the eligibility rules for your own packages.

Note. Don't confuse [package](#) eligibility rules with those associated with a [campaign](#). Package eligibility rules control whether a customer is allowed to choose a package on the order transaction. Campaign eligibility rules control whether a campaign is highlighted on the "eligible campaigns" [content zone](#) and on an order's [eligibility tree](#).

Contents

[Criteria Groups versus Eligibility Criteria](#)
[Defining Logical Criteria](#)
[Examples Of Package Eligibility Rules](#)

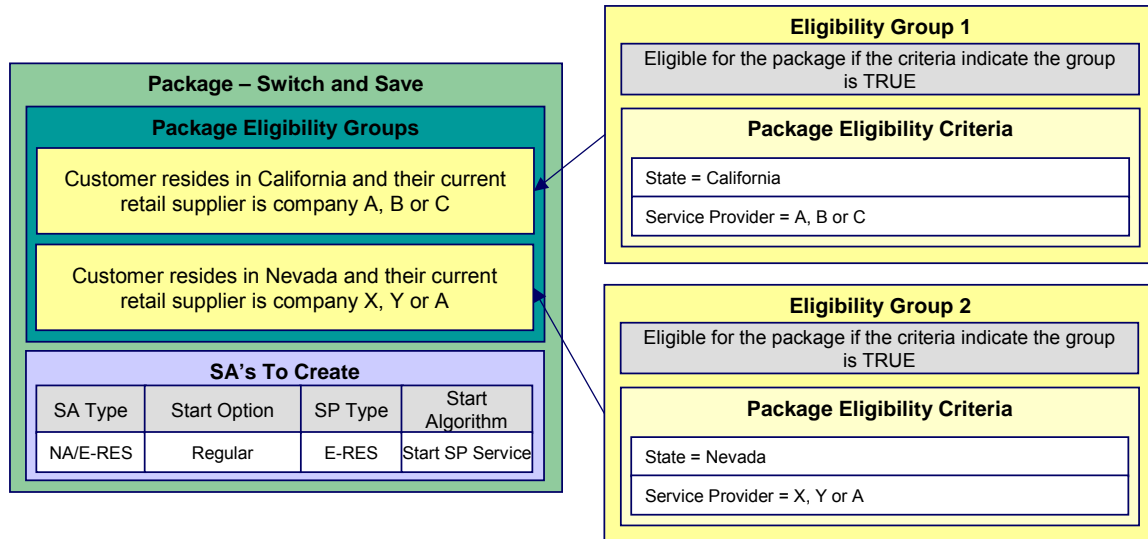
Criteria Groups versus Eligibility Criteria

Before we provide concrete examples of eligibility criteria, we need to explain two concepts: Criteria Groups and Eligibility Criteria. A package's criteria groups control whether a customer is eligible to choose a package. 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 for a given customer.
- When you create a group, you define what should happen if the group is TRUE or FALSE. You have the following choices:
 - The customer is eligible to choose the package
 - The customer is not eligible to choose the package
 - The next group should be checked.

We'll use the following example to help illustrate these points. Assume a package exists that can only be selected if:

- The customer resides in California and their current retail supplier is company A, B or C
- OR, the customer resides in Nevada and their current retail supplier is company X, Y or A



This package requires two eligibility groups because it has two distinct conditions:

- IF (STATE = CALIFORNIA AND (SERVICE PROVIDER IN (A, B, C)))
- IF (STATE = NEVADA AND (SERVICE PROVIDER IN (X, Y, A)))

If either condition is true, then the customer is eligible for the package.

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 resides in California and their current retail supplier is company A, B or C	<i>Eligible</i>	<i>Check next group</i>
2	Customer resides in Nevada and their current retail supplier is company X, Y or A	<i>Eligible</i>	<i>Ineligible</i>

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

Group 1: Customer resides in California and their current retail supplier is company A, B or C				
Seq	Logical Criteria	If Eligibility Criteria is TRUE	If Eligibility Criteria is FALSE	If Insufficient Data
10	State = California	<i>Check next condition</i>	<i>Group is false</i>	<i>Group is false</i>
20	Electric retail supplier in (A, B, C)	<i>Group is true</i>	<i>Group is false</i>	<i>Group is false</i>

Group 2: Customer resides in Nevada and their current retail supplier is company X, Y or A				
Seq	Logical Criteria	If Eligibility Criteria is TRUE	If Eligibility Criteria is FALSE	If Insufficient Data
10	State = Nevada	<i>Check next condition</i>	<i>Group is false</i>	<i>Group is false</i>

20	Electric retail supplier in (X, Y, A)	<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 choose one of the miscellaneous fields on the order (keep in mind that miscellaneous fields also hold the answers to any questions posed on the order).
- 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 whose job it will be to perform the comparison (and return 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 Package Eligibility Rules](#) provide numerous examples to help you understand this design.

Examples Of Package Eligibility Rules

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

Contents

- [A Package That Is Always Eligible](#)
- [A Package With A Time Span Comparison](#)
- [A Package With Service Type Comparison](#)
- [A Package With More Complex Operators](#)

A Package That Is Always Eligible

If a package is always eligible, it still needs an eligibility group. Good examples of such packages are shown under [An Easier Way To Create One Time Charges](#). A package that is always eligible would require the following eligibility information:

Group No.	Group Description	If Group is TRUE	If Group is FALSE
1	Always eligible	<i>Eligible</i>	<i>Eligible</i>

The following criteria will be required for this group:

Group 1: Always eligible					
Seq	Field to Compare	Comparison Method	If TRUE	If FALSE	If Insufficient Data
10	Algorithm: retrieve order's account	Algorithm: Always true	<i>Group is true</i>	<i>Group is true</i>	<i>Group is true</i>

This group's criterion is odd in that it's not comparing anything. In order to setup this criterion, you have to use the following "trick":

- Field to Compare. You can pick any miscellaneous field on the campaign. If the campaign has no miscellaneous fields, you'll need to setup an algorithm to retrieve a field from the order (refer to [PKEL-ENRFLD](#) for an example of this type of algorithm). In the above example, we used an algorithm to retrieve the order's account.
- Comparison Method. We chose a comparison algorithm that always returns a value of **TRUE** (refer to [PKCC-DEFAULT](#) for an example of this type of algorithm).
- You'll notice that if a value of **TRUE** is returned, the *Group is true* (and we've setup the group to indicate a true group means the customer is eligible for the package).

We understand this is a little complicated, but this design provides the power necessary to support very complex criteria. The following sections provide more examples to help you form a better understanding of these concepts.

A Package With A Time Span Comparison

The [Senior Citizen Package](#) has the following eligibility rules:

- Customer class = Residential
- Division = California
- Birth date equates to that of a senior citizen

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

Group No.	Group Description	If Group is TRUE	If Group is FALSE
1	Residential, Calif, Senior	<i>Eligible</i>	<i>Ineligible</i>

The following criteria will be required for this group:

Group 1: Residential, Calif, Senior					
Seq	Field to Compare	Comparison Method	If TRUE	If FALSE	If Insufficient Data
10	Algorithm: retrieve order's customer class	= R	<i>Check next condition</i>	<i>Group is false</i>	<i>Group is false</i>
20	Algorithm: retrieve order's division	= CA	<i>Check next condition</i>	<i>Group is false</i>	<i>Group is false</i>
30	Misc Field On Order: Date of Birth	Algorithm: True if senior	<i>Group is true</i>	<i>Group is false</i>	<i>Group is false</i>

The first two criteria are easy; they simply retrieve a field on the order and compare it to a given value. If the comparison doesn't result in a TRUE value, the **Group is false** (and, the group indicates that if the group is false, the customer isn't eligible for the package). If the comparison results in a TRUE value, the next condition is checked.

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

- **Field to Compare.** The miscellaneous field 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 [PKCC-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 customer is eligible).

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

A Package With Service Type Comparison

The [Has Gas / No Electricity Package](#) has the following eligibility rules:

- Customer has gas service
- Customer doesn't have electricity service

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

Group No.	Group Description	If Group is TRUE	If Group is FALSE
1	Has Gas, Doesn't Have Electricity	<i>Eligible</i>	<i>Ineligible</i>

The following criteria will be required for this group:

Group 1: Has Gas, Doesn't Have Electricity					
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 false</i>	<i>Group is true</i>	<i>Group is false</i>

Both criteria are similar – they check if the customer has a given type of service and act accordingly. We used a “field to compare” algorithm to pull this off, but this algorithm is a bit counter intuitive (but understanding it will provide you with another way to implement complex eligibility criteria):

- **Field to Compare.** We chose a “field to compare” algorithm that checks if an 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 [PKEL-SVCTYP](#) for an example 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 is not a misspelling). This value is, in turn, compared by the “comparison method” and the group is set to true, false or check next accordingly.

A Package With More Complex Operators

One of the eligibility groups described under [Criteria Groups versus Eligibility Criteria](#) has the following eligibility rules:

- Customer must live in California
- Service provider is A, B or C

The eligibility group in which these rules would be defined looks as follows:

Group No.	Group Description	If Group is TRUE	If Group is FALSE
1	Lives in California, current service provider is either retailer A, B or C	<i>Eligible</i>	<i>Ineligible</i>

The following criteria will be required for this group:

Group 1: Lives in California, current service provider is either retailer A, B or C					
Seq	Field to Compare	Comparison Method	If TRUE	If FALSE	If Insufficient Data
10	Algorithm: retrieve order's state	= CA	<i>Check next condition</i>	<i>Group is false</i>	<i>Group is false</i>
20	Misc Field On Order: Current electric supplier	IN A, B, C	<i>Group is true</i>	<i>Group is false</i>	<i>Group is false</i>

This example is very simple. The only thing worth pointing out is that a number of operators can be used when you define a “comparison method”. Refer to [Package - Eligibility](#) for the complete list of operators.

The Big Picture Of Campaign Eligibility Rules

In the previous section, we explained how to [Design Eligibility Rules For A Package](#). In this section, we explain how to design eligibility rules for a [campaign](#).

A campaign's eligibility rules control two things:

- Whether the campaign appears in the Eligible Campaigns [content zone](#). This content zone suggests possible campaigns for the current customer displayed on control central. You might want to use this zone if you have different campaigns that are offered to different types of customers. Refer to [Real-time Marketing of Additional Services to a Customer](#) for an example of how this content zone can be used to “up sell” additional services.
- Whether the campaign appears on an order's [eligibility tree](#). Campaigns that appear in this tree are alternate campaigns that may be used on the order. Refer to [Real-time Marketing of Services to a Prospect](#) for an example of how an order's eligibility tree can contain alternate campaigns for an order.

Designing these criteria can be easy or time-consuming; it all depends on the complexity of your business rules. We'll walk you through several examples to help you form an intuitive understanding of how to setup campaign eligibility rules. Once you've acquired this intuition you'll be ready to design the eligibility rules for your own packages.

Note. Don't confuse eligibility rules for a package with those associated with a campaign. Package eligibility rules control whether a customer is allowed to choose a package. Campaign eligibility rules control whether a campaign is highlighted on the Eligible Campaigns [content zone](#) and on an order's [eligibility tree](#).

Contents

[Campaign Eligibility Rules Are Not Strictly Enforced](#)
[Campaign Eligibility Rules Are Defined Using Algorithms](#)
[Examples Of Campaign Eligibility Rules](#)

Campaign Eligibility Rules Are Not Strictly Enforced

While the system prevents ineligible packages from being selected for an order (based on the [package's eligibility rules](#)), the system does not strictly enforce a campaign's eligibility rules. In other words, a user can change the campaign on an order to any **active** campaign. This “laxness” is intentional; otherwise it could become impossible to change an order's campaign.

It might be more helpful to think of campaign eligibility rules as “highlight conditions”. These “highlight conditions” simply control whether the campaign appears in the Eligible Campaigns [content zone](#) and on an order's [eligibility tree](#).

Campaign Eligibility Rules Are Defined Using Algorithms

You define a campaign's eligibility rules using one or more algorithms on the [campaign](#). These algorithms are relatively straightforward as they simply return one of the following values:

- **Eligible.** Meaning the customer is eligible for the campaign.

- **Ineligible.** Meaning the customer is not eligible for the campaign
- **Check Next Algorithm.** Meaning that the next algorithm should be checked. You'd only design an algorithm to return such a value if multiple algorithms are used to determine a customer's eligibility for a campaign.

Bottom line. A customer is eligible for a campaign if at least one of its eligibility algorithms returns a value of "eligible".

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

Examples Of Campaign Eligibility Rules

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

Contents

- [A Campaign That Is Always Eligible](#)
- [A Campaign That Is Never Eligible](#)
- [A Campaign For Residential Customers](#)
- [A Campaign For California, Residential Customers](#)

A Campaign That Is Always Eligible

If a campaign is always eligible, it would require a simple eligibility algorithm that returns an indication that it is always **eligible**. A good example of such a campaign is shown under [An Easier Way To Create One Time Charges](#). Refer to [CAEL-DEFAULT](#) for an example of this type of eligibility algorithm.

A Campaign That Is Never Eligible

If a campaign is never eligible, it would require a simple eligibility algorithm that returns an indication that it is **ineligible**. A good example of such a campaign is shown under [Prelude To Start / Stop](#). Refer to [CAEL-DEFAULT](#) for an example of this type of eligibility algorithm.

A Campaign For Residential Customers

If a campaign should only be targeted at residential customers, it would require a simple eligibility algorithm that returns an indication of **eligible** if the customer's customer class is residential. A good example of such a campaign is shown under [Marketing Surveys](#). Refer to [CAEL-CC](#) for an example of this type of eligibility algorithm.

A Campaign For California, Residential Customers

If a campaign should only be targeted at residential customers in California, it would require two eligibility algorithms:

Algorithm No	Algorithm	Example Algorithm
1	Indicate Check Next Algorithm if customer class is "residential"	Refer to CAEL-CC for an example of this type of eligibility algorithm.
2	Indicate Eligible if division is "California"	Refer to CAEL-DIV for an example of this type

		of eligibility algorithm.
--	--	---------------------------

Designing Campaigns and Packages

Designing [campaigns](#) and [packages](#) can be simple or challenging. Factors that will influence the complexity of the design process include:

- Whether the type of goods and services marketed to your customers / prospects differs depending on some trait of the customer or the region in which they are located. For example,
 - Your company may have very straightforward packages that only differ based on whether the customer is industrial, residential or commercial. If this describes your company, the setup process will be simple.
 - Alternatively, your company might offer different packages to residential customers based on whether they pay automatically, whether they are a senior citizen, whether they purchases multiple services from you, etc. If this describes your company, then you'll have to define the various eligibility criteria and the respective packages applicable to customers who satisfy each criterion.
- Whether premises exist prior to order taking. For example, a company that distributes a commodity has a well-defined geographic area and therefore new premises typically don't arise when orders are taken.
- Whether your company offers many different types of goods and services.
- Whether your rates have charges / discounts that are only applied to some customers. For example, you might have discounts that are applied if the customer is a senior citizen or an employee.
- Which of the [Supported Business Processes](#) you implement.

The topics that follow provide a structured approach to design your campaigns and packages.

Contents

- Determine The Place Of Start / Stop
- Determine Which Types Of Service Agreements Are Created
- Determine The Different Start Options That Exist For Each Type Of Service Agreement
- Determine The SP Type (if any) For Each Service Agreement
- Determine The Method Used To Create Each Service Agreement
- Extract Algorithms From The Start Method
- Determine Which Combinations Can Be Grouped Under A Package
- Determine Package Eligibility Rules
- Extract Algorithms From The Eligibility Rules
- Extract Miscellaneous Fields From The Eligibility Rules
- Determine Additional Information That Is Required If A Package Is Selected
- Determine If There Are Additional Questions / Fields For Your Packages
- Determine If Additional Instructions Should Be Shown
- Determine If There Are Campaign-Level Fields / Questions
- Determine The Properties Of Every Miscellaneous Field
- Extract Characteristic From The Miscellaneous Fields
- Extract Column References From The Miscellaneous Fields
- Extract Algorithms From The Column References
- Determine Account / Premise Dependency For Each Column Reference
- Determine If Any Properties Are Overridden On Any Package

Determine The Eligibility Rules For The Campaign
 Determine How The Campaign Behaves
 Setup Sequence

Determine The Place Of Start / Stop

Before you can design your campaigns and packages, you must determine if your organization will use [start / stop](#) to initiate new service agreements or whether new service can be initiated using the [order](#) transaction. Refer to [Order versus Start / Stop](#) for information to help you make this decision.

If you decide to use [start / stop](#) to initiate service, we'd recommend creating the following types of campaigns:

- Create a campaign similar to that described under [Prelude to Start / Stop](#).
- If you levy one-time charges, you may also want to create a campaign similar to that described under [An Easier Way To Create One Time Charges](#).
- If your organization performs marketing surveys, you may also want to create a campaign similar to that described under [Marketing Surveys](#).

If you decide to use the [order](#) transaction to initiate new service, follow the steps outlined in the following sections. Note that we use the example described under [Packages Limited By Answers And Field Values](#) when describing the design process.

Determine Which Types Of Service Agreements Are Created

Assumption! We have assumed that you have already designed your organization's SP types, SA types, start options and rates. If you have not yet done this, we recommend familiarizing yourself with the concepts described in [Designing SP Types](#), [Rates](#) and [Defining Service Agreement Types](#) before designing your campaigns and packages.

Start off by determining the different types of service agreements that can be created if a package is selected. In our [example campaign](#), only one SA type is used – **E-RES** (electric residential service).

Determine The Different Start Options That Exist For Each Type Of Service Agreement

After you know which SA types are going to be created, determine the potential [start options](#) that can be used for each SA type. In our [example campaign](#), several start options are possible for the **E-RES** (electric residential service):

SA Type	Start Option
<i>E-RES</i>	<i>Senior</i>
<i>E-RES</i>	<i>Single-Elec Heat</i>
<i>E-RES</i>	<i>Single-Gas Heat</i>
<i>E-RES</i>	<i>Multi-Family</i>

Determine The SP Type (if any) For Each Service Agreement

The SA types and start options control how the service agreement will look (i.e., they control the rate, contract riders, characteristics, interval profiles, time-of-use maps, etc. defaulted onto new service agreements created when a package is selected). During this step, you define the SP type of the service point that will be linked to the service agreement. If several different types of SP types can be used for a service agreement, you'll have to choose one as this is the type of service point that will be created when the service agreement is created. If you plan to allow the package to be used to start service at an existing premise (and you don't want the system to create new service points), you will have to develop new plug-ins. Please speak to your implementation group if you need more information.

In our [example campaign](#), only one SP type is created (or reused if the premise already has a service point of this type) – **E-RES** (electric residential service point):

SA Type	Start Option	SP Type
<i>E-RES</i>	<i>Senior</i>	<i>E-RES</i>
<i>E-RES</i>	<i>Single-Elec Heat</i>	<i>E-RES</i>
<i>E-RES</i>	<i>Single-Gas Heat</i>	<i>E-RES</i>
<i>E-RES</i>	<i>Multi-Family</i>	<i>E-RES</i>

Many service agreements don't need SP types. For service agreements that don't use service points (e.g., deposit service agreements, charitable contributions, one-time charges, etc.), you don't have to reference a SP type.

Determine The Method Used To Create Each Service Agreement

For each SA Type / Start Option / SP Type combination, determine the algorithm that will be used to create the service agreement and the service point. The base package is supplied with a limited number of such algorithm types (refer to [STRM-AS](#) and [STRM-VT](#) for examples).

SA Type	Start Option	SP Type	Start Method
E-RES	Senior	E-RES	Algorithm: Create new service agreement and stop old service agreement if active SA with the same service type exists
E-RES	Single-Elec Heat	E-RES	Algorithm: Create new service agreement and stop old service agreement if active SA with the same service type exists
E-RES	Single-Gas Heat	E-RES	Algorithm: Create new service agreement and stop old service agreement if active SA with the same service type exists
E-RES	Multi-Family	E-RES	Algorithm: Create new service agreement and stop old service agreement if active SA with the same service type exists

Extract Algorithms From The Start Method

The start methods illustrated above all require the same algorithm:

Algorithm Type	Algorithm	Parameter Values
Create SP and create SA (see STRM-VT)	Always create new SA and stop existing SA	2 (stop existing service)

Determine Which Combinations Can Be Grouped Under A Package

Now you determine which combinations of service agreements can be offered to a customer as a package. In our [example campaign](#), a customer can only choose one type of service agreement (i.e., each entry in our table is mutually exclusive). This means that each entry is a separate package:

Package	SA Type	Start Option	SP Type	Algorithm
1	<i>E-RES</i>	<i>Senior</i>	<i>E-RES</i>	<i>STRM-VT (Switch Option)</i>
2	<i>E-RES</i>	<i>Single-Elec Heat</i>	<i>E-RES</i>	<i>STRM-VT (Switch Option)</i>
3	<i>E-RES</i>	<i>Single-Gas Heat</i>	<i>E-RES</i>	<i>STRM-VT (Switch Option)</i>
4	<i>E-RES</i>	<i>Multi-Family</i>	<i>E-RES</i>	<i>STRM-VT (Switch Option)</i>

Refer to [Packages Limited By Current State Of Service](#) for an example of a package that has multiple service agreements.

Determine Package Eligibility Rules

The topics in this section describe the eligibility rules for each package in our [example campaign](#).

Assumption. This section assumes you are comfortable with the information in [The Big Picture Of Package Eligibility](#).

Contents

- [Eligibility Rules For Package 1](#)
- [Eligibility Rules For Package 2](#)
- [Eligibility Rules For Package 3](#)
- [Eligibility Rules For Package 4](#)

Eligibility Rules For Package 1

Package 1 (the senior package) in our [example campaign](#) has the following eligibility rules:

- Customer class = Residential
- Division = California
- Birth date equates to that of a senior citizen

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

Group No.	Group Description	If Group is TRUE	If Group is FALSE
1	Residential, Calif, Senior	<i>Eligible</i>	<i>Ineligible</i>

The following criteria will be required for this group:

Group 1: Residential, Calif, Senior					
Seq	Field to Compare	Comparison Method	If TRUE	If FALSE	If Insufficient Data
10	Algorithm: retrieve order's customer class	= R	<i>Check next condition</i>	<i>Group is false</i>	<i>Group is false</i>
20	Algorithm: retrieve order's division	= CA	<i>Check next condition</i>	<i>Group is false</i>	<i>Group is false</i>
30	Misc Field On Order: Date of Birth	Algorithm: True if senior	<i>Group is true</i>	<i>Group is false</i>	<i>Group is false</i>

Eligibility Rules For Package 2

Package 2 (the single-family / electric heat package) in our [example campaign](#) has the following eligibility rules:

- Customer class = Residential
- Division = California
- Birth date does not equate to that of a senior citizen
- Lives in a single-family home
- House is heated with electricity

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

Group No.	Group Description	If Group is TRUE	If Group is FALSE
1	Residential, Calif, Non Senior, Single Family, Electric Heat	<i>Eligible</i>	<i>Ineligible</i>

The following criteria will be required for this group:

Group 1: Residential, Calif, Non Senior, Single Family, Electric Heat					
Seq	Field to Compare	Comparison Method	If TRUE	If FALSE	If Insufficient Data
10	Algorithm: retrieve order's customer class	= R	<i>Check next condition</i>	<i>Group is false</i>	<i>Group is false</i>
20	Algorithm: retrieve order's division	= CA	<i>Check next condition</i>	<i>Group is false</i>	<i>Group is false</i>
30	Misc Field On Order: Date	Algorithm: True if senior	<i>Group is</i>	<i>Check next</i>	<i>Group is false</i>

	of Birth		<i>false</i>	<i>condition</i>	
40	Misc Field On Order: Type of Residence	= S (single family home)	<i>Check next condition</i>	<i>Group is false</i>	<i>Group is false</i>
50	Misc Field On Order: Type of Heat	= E (electric)	<i>Group is true</i>	<i>Group is false</i>	<i>Group is false</i>

Eligibility Rules For Package 3

Package 3 (the single-family / gas heat package) in our [example campaign](#) has the following eligibility rules:

- Customer class = Residential
- Division = California
- Birth date does not equate to that of a senior citizen
- Lives in a single-family home
- House is heated with gas

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

Group No.	Group Description	If Group is TRUE	If Group is FALSE
1	Residential, Calif, Non Senior, Single Family, Gas Heat	<i>Eligible</i>	<i>Ineligible</i>

The following criteria will be required for this group:

Group 1: Residential, Calif, Non Senior, Single Family, Gas Heat					
Seq	Field to Compare	Comparison Method	If TRUE	If FALSE	If Insufficient Data
10	Algorithm: retrieve order's customer class	= R	<i>Check next condition</i>	<i>Group is false</i>	<i>Group is false</i>
20	Algorithm: retrieve order's division	= CA	<i>Check next condition</i>	<i>Group is false</i>	<i>Group is false</i>
30	Misc Field On Order: Date of Birth	Algorithm: True if senior	<i>Group is false</i>	<i>Check next condition</i>	<i>Group is false</i>
40	Misc Field On Order: Type of Residence	= S (single family home)	<i>Check next condition</i>	<i>Group is false</i>	<i>Group is false</i>
50	Misc Field On Order: Type of Heat	= G (gas)	<i>Group is true</i>	<i>Group is false</i>	<i>Group is false</i>

Eligibility Rules For Package 4

Package 4 (the multi-family package) in our [example campaign](#) has the following eligibility rules:

- Customer class = Residential

- Division = California
- Birth date does not equate to that of a senior citizen
- Lives in a multi-family home

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

Group No.	Group Description	If Group is TRUE	If Group is FALSE
1	Residential, Calif, Non Senior, Multi Family	<i>Eligible</i>	<i>Ineligible</i>

The following criteria will be required for this group:

Group 1: Residential, Calif, Non Senior, Single Family, Gas Heat					
Seq	Field to Compare	Comparison Method	If TRUE	If FALSE	If Insufficient Data
10	Algorithm: retrieve order's customer class	= R	<i>Check next condition</i>	<i>Group is false</i>	<i>Group is false</i>
20	Algorithm: retrieve order's division	= CA	<i>Check next condition</i>	<i>Group is false</i>	<i>Group is false</i>
30	Misc Field On Order: Date of Birth	Algorithm: True if senior	<i>Group is false</i>	<i>Check next condition</i>	<i>Group is false</i>
40	Misc Field On Order: Type of Residence	= M (multi family home)	<i>Group is true</i>	<i>Group is false</i>	<i>Group is false</i>

Extract Algorithms From The Eligibility Rules

The eligibility rules illustrated above require the following algorithms:

Algorithm Type	Algorithm	Parameter Values
Retrieve a field from the order for eligibility comparison (see PKEL-ENRFLD)	Retrieve Order's Customer Class	Field name: CUST_CL_CD
Retrieve a field from the order for eligibility comparison (see PKEL-ENRFLD)	Retrieve Order's Division	Field name: CIS_DIVISION
Time span comparison (see PKCC-TIMESPN)	Determine if birth date equates to that of a senior	Time span type: YEAR Operator: >= Comparison Value: 65

Extract Miscellaneous Fields From The Eligibility Rules

During this step, you extract miscellaneous fields that must be captured on the order in order to drive your eligibility rules.

Misc. Field	Where Stored	Why Is It Needed
Date of birth	Person characteristic	Numerous eligibility rules use this field to determine if a customer is a senior citizen

Type of Residence	Premise characteristic	Numerous eligibility rules use this field to qualify a customer for a package
Type of Heat	Premise characteristic	Numerous eligibility rules use this field to qualify a customer for a package

Fields explicitly referenced on every order. You'll notice that we didn't list those fields referenced in the eligibility rules that are explicitly referenced on every order (e.g., customer class, state, division, etc.). This is because these types of fields come "for free".

Determine Additional Information That Is Required If A Package Is Selected

During this step, you determine if there's any information that needs to be created when a package is selected that can't be defaulted from its start option. In our [example campaign](#), package 4 (the Multi-Family Home package), also needs to know the number of housing units at the premise (this information is probably needed to calculate a specific charge in the rate). None of the other packages require additional information. The extra field has been appended to our misc. field table.

Misc. Field	Where Stored	Why Is It Needed
Date of birth	Person characteristic	Numerous eligibility rules use this field to determine if a customer is a senior citizen
Type of Residence	Premise characteristic	Numerous eligibility rules use this field to qualify a customer for a package
Type of Heat	Premise characteristic	Numerous eligibility rules use this field to qualify a customer for a package
Number of Units in House	Premise characteristic	The electric SA created by package 4 requires the number of units when it calculates the ongoing bill segments

Determine If There Are Additional Questions / Fields For Your Packages

Our [example campaign](#) indicates that all customers are encouraged to pay automatically. In order to pay automatically several fields are required (Start Date, End Date, Autopay Source (i.e., bank / credit card), Account Number, Account Name, Expiration Date). These fields are not "explicitly referenced" on the order and therefore must be captured as miscellaneous fields. We have appended these fields to our list of miscellaneous fields:

Misc. Field	Where Stored	Why Is It Needed
Date of birth	Person characteristic	Numerous eligibility rules use this field to determine if a customer is a senior citizen
Type of Residence	Premise characteristic	Numerous eligibility rules use this field to qualify a customer for a package
Type of Heat	Premise characteristic	Numerous eligibility rules use this field to qualify a customer for a package
Number of Units in	Premise characteristic	The electric SA created by package 4 requires the number of

House		units when it calculates the ongoing bill segments
Auto Pay Start Date	Account / Autopay Start Date Column	Used if a customer pays automatically
Auto Pay End Date	Account / Autopay End Date Column	Used if a customer pays automatically
Auto Pay Source	Account / Autopay Source	Used if a customer pays automatically
Auto Pay Account Number	Account / Autopay Account ID	Used if a customer pays automatically
Auto Pay Account Name	Account / Autopay Account Name	Used if a customer pays automatically
Auto Pay Account Expiration Date	Account / Autopay Account Expiration Date	Used if a customer pays automatically

We're going to assume that we don't capture any additional marketing-oriented questions. If we did, we'd need additional fields to hold the response to each question.

Determine If Additional Instructions Should Be Shown

If a package is selected, you can present additional instructions to the user informing them of special tasks they should perform (or special instructions they should give to the customer). For example, you could instruct the user to inform customers who aren't senior citizens that they should call back if they become seniors as they may qualify for a better rate. These instructions can be defined on each package.

For the sake of brevity, we will assume no additional instructions are necessary when a package is selected.

Determine If There Are Campaign-Level Fields / Questions

Next, you determine if there are additional fields that need to be captured regardless of the package that's selected. For example, you may want to capture additional demographic or geographic information for marketing purposes.

While our [example campaign](#) has no such requirement, for illustration purposes we'll assume you ask the customer to rate the quality of service they have experienced (assuming they are a current customer). This additional question requires another entry in our list of miscellaneous fields.

Misc. Field	Where Stored	Why Is It Needed
Date of birth	Person characteristic	Numerous eligibility rules use this field to determine if a customer is a senior citizen
Type of Residence	Premise characteristic	Numerous eligibility rules use this field to qualify a customer for a package
Type of Heat	Premise characteristic	Numerous eligibility rules use this field to qualify a customer for a package

Number of Units in House	Premise characteristic	The electric SA created by package 4 requires the number of units when it calculates the ongoing bill segments
Auto Pay Start Date	Account / Autopay Start Date Column	Used if a customer pays automatically
Auto Pay End Date	Account / Autopay End Date Column	Used if a customer pays automatically
Auto Pay Source	Account / Autopay Source	Used if a customer pays automatically
Auto Pay Account Number	Account / Autopay Account ID	Used if a customer pays automatically
Auto Pay Account Name	Account / Autopay Account Name	Used if a customer pays automatically
Auto Pay Account Expiration Date	Account / Autopay Account Expiration Date	Used if a customer pays automatically
Service Rating	Person characteristic	Campaign level marketing question

Determine The Properties Of Every Miscellaneous Field

At this point, you have amalgamated all miscellaneous fields required for the campaign and its packages. Now you have to define the following for each field:

- Prompt On Order. This is the prompt that will prefix the field when the [order](#) is taken. The following table provides examples.
- Campaign Applicability. This controls whether the field is **optional**, **required**, or **only applicable on package(s)**.
 - A value of **required** means the order taker must enter a valid value before packages are displayed for selection.
 - A value of **optional** means the order taker can leave the field blank prior to package selection. You can make an **optional** field **required** on individual packages. For example, you would probably make **Date of birth** required on the **Senior** package.
 - A value of **only applicable on packages** is used for fields that are only captured for specific packages. Fields of this type are suppressed prior to package selection. When you setup a package, you indicate if the field is applicable (if so, the user is only prompted for the field if the package is selected).
- Type Of Response. This controls where the field is stored when the order is completed and how it is validated. You have two options: **characteristic** or **column**.
 - If a field is a **Characteristic**, you must define which entity the characteristic value is to be stored on when the order is completed (**Person**, **Account**, **Premise** or **Order**). You must also define the **Characteristic Type** as this controls the characteristic value entered by the user.
 - If a field's value resides is a **column** (as opposed to a characteristic value), you must define the [column reference](#). A field's column reference controls three things:

- How the field's value is retrieved. A field's value is retrieved when an existing person / account / premise is referenced on an order.
- How the field's value is validated. A field's value is validated before package eligibility is determined and before the order is completed.
- How the field's value is posted (i.e., updated on the database). A field's value is posted when an order is completed.

Refer to [Extract Column References From Miscellaneous Fields](#) for more information about column references.

- **Default Value On Order.** This controls whether the order taker is supplied with a default value when an order is created. Note well: for **optional** and **only applicable on package** fields, you can specify a different default value on each package.

Misc. Field	Prompt On Order	Campaign Applicability	Type Of Response	Default Value On Order
Date of birth	What is your date of birth (used to qualify for senior discount)?	<i>Optional</i>	<i>Characteristic</i> Entity: <i>Person</i> Char Type: Date of birth	
Type of Residence	What type of residence do you live in?	<i>Required</i>	<i>Characteristic</i> Entity: <i>Premise</i> Char Type: Type of residence	S (single family)
Type of Heat	How do you heat your home?	<i>Required</i>	<i>Characteristic</i> Entity: <i>Premise</i> Char Type: Type of heat	E (electric)
Number of Units in House	How many units are in the entire house?	<i>Only applicable on package(s)</i>	<i>Characteristic</i> Entity: <i>Premise</i> Char Type: Number of units in house	
Auto Pay Start Date	What day would you like to start automatic payment processing (YYYY-MM-DD)?	<i>Optional</i>	<i>Column</i> Column Reference: Auto Pay Start Date	
Auto Pay End Date	What day would you like to stop automatic payment processing (leave blank if not applicable)?	<i>Optional</i>	<i>Column</i> Column Reference: Auto Pay End Date	
Auto Pay Source	From what type of credit card / bank will the funds be debited?	<i>Optional</i>	<i>Column</i> Column Reference: Auto Pay Source	
Auto Pay Account	What is your credit card / bank account number?	<i>Optional</i>	<i>Column</i> Column Reference:	

Number			Auto Pay Account Number	
Auto Pay Account Name	What is the name on your account?	<i>Optional</i>	<i>Column</i> Column Reference: Auto Pay Account Name	
Auto Pay Account Expiration Date	If paying with a credit card, what is the expiration date (MM-YYYY)?	<i>Optional</i>	<i>Column</i> Column Reference: Auto Pay Credit Card Expiration Date	
Service Rating	Please rate our service	<i>Optional</i>	<i>Characteristic</i> Entity: <i>Person</i> Char Type: Service rating	1 (excellent)

Extract Characteristic From The Miscellaneous Fields

Characteristic Type	Type	Valid Values	Entities On Which Characteristic Can Be Defined
Date of birth	<i>Ad hoc value</i> : use validation algorithm to confirm date is > 1-Jan-1900 and < 31-Feb-2002	Not applicable	<i>Person, Order</i>
Type of Residence	<i>Predefined value</i>	S: Single-Family M: Multi-Family	<i>Premise, Order</i>
Type of Heat	<i>Predefined value</i>	E: Electric G: Gas	<i>Premise, Order</i>
Number of Units in House	<i>Ad hoc value</i> : use validation algorithm to confirm value is > 0	Not applicable	<i>Premise, Order</i>
Service Rating	<i>Predefined value</i>	0: Not applicable 1: Excellent 2: Very Good 3: Good 4: Poor	<i>Person, Order</i>

Extract Column References From The Miscellaneous Fields

The [miscellaneous fields](#) illustrated above require the following column references (we describe the referenced algorithms at the bottom of the table):

Column Reference	Retrieval Algorithm	Validation Algorithm	Post Algorithm	Foreign Key Reference
------------------	---------------------	----------------------	----------------	-----------------------

Auto Pay Start Date	Retrieve account's current automatic payment information	Validate automatic payment information	Post automatic payment information	
Auto Pay End Date	N/A – this field is handled by the Retrieval Algorithm on Auto Pay Start Date	N/A – this field is handled by the Validation Algorithm on Auto Pay Start Date	N/A – this field is handled by the Post Algorithm on Auto Pay Start Date	
Auto Pay Source	N/A – this field is handled by the Retrieval Algorithm on Auto Pay Start Date	N/A – this field is handled by the Validation Algorithm on Auto Pay Start Date	N/A – this field is handled by the Post Algorithm on Auto Pay Start Date	Auto pay source
Auto Pay Account Number	N/A – this field is handled by the Retrieval Algorithm on Auto Pay Start Date	N/A – this field is handled by the Validation Algorithm on Auto Pay Start Date	N/A – this field is handled by the Post Algorithm on Auto Pay Start Date	
Auto Pay Account Name	N/A – this field is handled by the Retrieval Algorithm on Auto Pay Start Date	N/A – this field is handled by the Validation Algorithm on Auto Pay Start Date	N/A – this field is handled by the Post Algorithm on Auto Pay Start Date	
Auto Pay Expiration Date	N/A – this field is handled by the Retrieval Algorithm on Auto Pay Start Date	N/A – this field is handled by the Validation Algorithm on Auto Pay Start Date	N/A – this field is handled by the Post Algorithm on Auto Pay Start Date	

There are three types of algorithms referenced in the above table:

- **Retrieval algorithm.** A column's retrieval algorithm retrieves the column's current value from the database. You'll notice that only one of our columns uses a retrieval algorithm. This is because we've used a sophisticated example of a group of columns that are retrieved at the same time. When you have such a group of columns, nominate one of the columns to do the retrieval work and specify the retrieval algorithm on it.
- **Validation algorithm.** A column's validation algorithm validates the column's value prior to the database being updated. You'll notice that only one of our columns uses a validation algorithm. This is because we've used a sophisticated example of a group of columns that are validated as a group. When you have such a group of columns, nominate one of the columns to do the validation work and specify the validation algorithm on it.
- **Post algorithm.** A column's post algorithm updates the column's value on the database when the order is completed. You'll notice that only one of our columns uses a post algorithm. This is because we've used a sophisticated example of a group of columns that are posted together. When you have such a group of columns, nominate one of the columns to do the posting work and specify the posting algorithm on it.

Notice that we have indicated that one of the columns has a “foreign key reference”. You declare a “foreign key reference” if the valid values for a column are defined on another table in the system (this also affects the order user interface as a dropdown appears for columns that have foreign key references). In our example, Auto Pay Source is the only column whose valid values are defined on the automatic payment source control table.

Extract Algorithms From The Column References

The column references illustrated above require the following algorithms:

Algorithm Type	Algorithm	Parameter Values
Retrieve automatic payment information from an account (see CRRT-AP)	Retrieve automatic payment information from an account	Param 1 = Auto Pay Start Date Column Reference Param 2 = Auto Pay End Date Column Reference Param 3 = Auto Pay Source Column Reference Param 4 = Auto Pay External Account Id Column Reference Param 5 = Auto Pay Expiration Date Column Reference Param 6 = Auto Pay Name Column Reference
Validate automatic payment information (see CRVL-AP)	Validate automatic payment information	Param 1 = Auto Pay Start Date Column Reference Param 2 = Auto Pay End Date Column Reference Param 3 = Auto Pay Source Column Reference Param 4 = Auto Pay External Account Id Column Reference Param 5 = Auto Pay Expiration Date Column Reference Param 6 = Auto Pay Name Column Reference
Post automatic payment information (see CRPS-AP)	Post automatic payment information	Param 1 = Auto Pay Start Date Column Reference Param 2 = Auto Pay End Date Column Reference Param 3 = Auto Pay Source Column Reference Param 4 = Auto Pay External Account Id Column Reference Param 5 = Auto Pay Expiration Date Column Reference Param 6 = Auto Pay Name Column Reference

Determine Account / Premise Dependency For Each Column Reference

Later in the design process (see [Determine How The Campaign Behaves](#)), you define on the campaign whether an account (and premise) is required, optional or not allowed on its orders. Your selection impacts how you setup column references. This is a little complicated so let's use an example:

- Assume you've setup a campaign to indicate a premise can optionally be defined on its orders.
- If the order-taker doesn't define a premise on an order, any premise-oriented questions and miscellaneous fields (Q&MF) should be suppressed on the order.

- It's easy for the system to know which Q&MFs to suppress when a Q&MF's response is held in a characteristic (because you must define on the campaign if the response will reside in an account, premise, person or order characteristic). For column reference Q&MFs, the system doesn't know if the response is account-oriented, premise-oriented or neither because algorithms handle the validation / updating of the response.
- Therefore, when you use column references on a campaigns Q&MFs, you must define if the responses are account-oriented or premise-oriented (so that the system can suppress the Q&MF when the user doesn't enter an account or premise on the order).

The following table contains the column references defined above as well as their dependency

Column Reference	Question Dependency
Auto Pay Start Date	Must have account
Auto Pay End Date	Must have account
Auto Pay Source	Must have account
Auto Pay Account Number	Must have account
Auto Pay Account Name	Must have account
Auto Pay Expiration Date	Must have account

You'll notice that each column reference is account-oriented (this is because automatic payment fields are stored on the respective account).

Determine If Any Properties Are Overridden On Any Package

At this point, you have defined the properties of all miscellaneous fields at the campaign level. You now need to determine if any of your packages need to override any of these properties. In general, you can only override miscellaneous fields whose campaign applicability is **optional** or **only applicable on package(s)**.

Our [example campaign](#) has only two overrides:

Package 1 requires the following miscellaneous field overrides:

Misc. Field	Override Prompt (seen if a package is selected)	Override Package Applicability	Override Default Value
Date of birth	<i>Not overridden</i>	<i>Required</i>	<i>Not overridden</i>

Package 4 requires the following miscellaneous field overrides:

Misc. Field	Override Prompt (seen if a package is selected)	Override Package Applicability	Override Default Value
Number of Units in House	<i>Not overridden</i>	<i>Required</i>	2

Warning! If you override field properties on a campaign's packages, you will not be able to change the properties of the field at the campaign level.

Determine The Eligibility Rules For The Campaign

Assumption. This section assumes you are comfortable with the information in [The Big Picture Of Campaign Eligibility](#).

The easiest way to determine a campaign's eligibility is to examine each package's eligibility criteria and abstract common eligibility restrictions onto the campaign. It's important to remember that campaign-level eligibility is only used to highlight if the campaign MAY contain applicable packages; campaign-level eligibility is not strictly enforced.

The packages on our [example campaign](#) are all targeted at residential customers in California. Therefore we'll take advantage of two base package algorithms to construct the campaign's eligibility criteria.

Algorithm Type	Algorithm	Parameter Values
Check if account's customer class is in a predefined list (see CAEL-CC)	If customer class is "residential", check next algorithm	If customer's customer class is in list: 30 (check next) If customer's customer class is not in list: 20 (ineligible) List of customer classes: R
Check if account's division is in a predefined list (see CAEL-DIV)	If division is "California", customer is eligible	If customer's CIS division is in list: 10 (eligible) If customer's CIS division is not in list: 20 (ineligible) List of customer classes: CA

Determine How The Campaign Behaves

The final step before you're ready to setup the campaign involves defining a variety of behavioral functions as described below:

- You need to define how the campaign uses accounts and premises. Specifically, on a campaign you define if an account / premise is required / optional / not allowed on its orders. In addition, if an account / premise is required or optional, you can control whether new accounts / premises can be created when an order is completed (the alternative is to force each order to use an existing account / premise). These controls prevent the unwanted proliferation of new accounts and premises for campaigns that are targeted at existing accounts and premises.
- Many fields on an order (e.g., phone type, customer class, division, account management group) can be defaulted on orders created for new customers. You will see these fields on [Order - Main](#). Think about each such field and whether a default value is appropriate. Keep in mind that the order taker can override any of these default values.
- An order taker can indicate that they want to [hold an order](#) if they need to do further research before continuing with an order. A background process (referred to by the batch control ID **TD-ECBK**) will create a [To Do entry](#) for held orders as a reminder to get back in touch with the customer on a future date. You can define the default To Do role assigned to such To Do entries when you create a campaign. Keep in mind that this value can be overridden when the order taker holds an order.
- You can indicate on a campaign the transaction to which the user should be transferred when orders are completed. Refer to [Supported Business Processes](#) for examples of how you might want to use this field.

Setup Sequence

And now you're ready to set up your campaign. The following points describe the order in which this should be done:

Assumption! We have assumed that you have already designed your organization's SP types, SA types, start options and rates. If you have not yet done this, we recommend familiarizing yourself with the concepts described in [Designing SP Types](#), [Rates](#) and [Defining Service Agreement Types](#) before designing your campaigns and packages.

- Setup any new [column references](#) required by the campaign's [miscellaneous fields](#). You will have to leave the retrieve, validate and post algorithms blank until after the next step.
- Setup any new [algorithms](#) required by:
 - The packages' [eligibility rules](#).
 - The packages' [start methods](#).
 - The campaign's [eligibility rules](#).
 - The campaign's [column references](#).
- Update the relevant [column references](#) with the appropriate retrieve, validate and post algorithms.
- Setup any new [characteristic types](#) required by the campaign's [miscellaneous fields](#).
- Create the [campaign](#) using the information you designed above. Specifically:
 - You defined the question and miscellaneous fields under [Determine The Properties Of Every Miscellaneous Field](#).
 - You defined the eligibility rules under [Determine The Eligibility Rules For The Campaign](#).
 - You defined everything else under [Determine How The Campaign Behaves](#).
 - We recommend leaving the campaign's status as **Inactive** until all packages are entered.
- Create the campaign's [packages](#) using the information you designed above. Specifically:
 - You defined the question and miscellaneous field overrides under [Determine If Any Properties Are Overridden On A Package](#).
 - You defined the eligibility rules under [Determine Package Eligibility Rules](#).
 - You defined the instructions under [Determine If Additional Instructions Are Required](#).
- Change the [campaign's](#) status to **Active**.

The default campaign on the installation record. If the campaign you've setup is the "default" campaign used when a user presses the Add Order button on [Control Central](#), make sure to specify it as the default campaign on the [installation record](#). Refer to [Setting up a New Customer Prior To Starting Service](#) for more information.

Other Useful Information

The topics in this section contain miscellaneous information about the sales & marketing functions.

Contents

- [Order versus Start / Stop](#)
- [Orders Cannot Be Changed After They Are Completed](#)
- [Premise and SP Characteristics May Be Populated From Postal Defaults](#)
- [To Do List Processing](#)
- [Hold and Cancel Reason Codes](#)
- [Additional Things Can Happen When An Order Is Completed](#)
- [New Premises and Accounts Are Only Created if an Order is Completed](#)
- [Premise Geographic Types Are Populated In An Unusual Way](#)
- [Account And Premise Usage Are Controlled By The Campaign](#)

Order versus Start / Stop

There are two different ways to start service for a customer:

- As described under [definitions](#), you can use the [order](#) transaction to start service.
- Alternatively, you can use [start / stop](#).

The following table provides insight into the pros and cons of each method (after this table you will find our recommendations):

Issue	Order	Start / Stop
New customer setup	Information about the new customer can be entered on the order page. A person / account will be setup when the order is completed.	A new person and account must be setup prior to using start / stop. See How To Add A New Customer From Control Central for more information.
New premise setup	Information about the new premise can be entered on the order page. A premise (and service points) will be setup when the order is completed.	A new premise and service point(s) must be setup prior to using start / stop.
Updates to existing customer information	Changes to a customer's existing account and/or person info may be entered on the order page. The related person and/or account will be updated when the order is completed.	Start / stop cannot be used to update person or account information. Rather, you must transfer to the person or account page and make the necessary changes.
Incomplete orders	The order transaction allows you to record customer information and hold it without starting service. The customer can call back in the future and you can continue where you left off.	You cannot leave a start half-finished – you either have to complete the start or cancel it.
Stop current service, start new service	The order transaction does not currently provide a mechanism for stopping service (it can only be used	Start / stop can be used to stop a customer's current service and start new service.

	to start service).	
Automatic versus manual start option selection (start options are used to default a great deal of information onto a service agreement)	If a customer selects a package, the system creates service agreements using the start options defined in the package (i.e., you do not have to select a start option for each service agreement that's created).	When you start service, you must define the specific start option for each service agreement created (actually, you only have to do this for those types of service agreements that use start options).
Starting premise and non-premise oriented services	If a customer selects a package, the system can create both premise and non-premise oriented service agreements. For example, when a package is selected, the system can create electric, gas, deposit, and charitable contribution SA's (based on the information defined on the package).	Multiple premise-oriented service agreements can be created in one interaction. However, a separate interaction is required for each non-premise oriented service agreement.
Changes after completion	An order cannot be modified after it is completed. Any changes (cancellations prior to starting service, changes of the start date, etc.) must be performed using the start / stop transaction.	The start / stop transaction allows you to cancel and make changes to pending start service agreements.
Field activity visibility	After an order is completed, the user is transferred to control central where field activities associated with the premise can be seen in trees and context zones.	All field activities associated with all premises associated with all pending starts / stops are shown on the start / stop transaction.
Address changes	Address changes entered on the order transaction are applied when an order is completed (i.e., when a package is selected). They are not held awaiting service agreement activation.	When you enter an address change on the start / stop transaction, this address change is saved until the first pending start or stop is activated. This means that the address change is effectively held until the start / stop is activated.

Based on the above, we'd recommend the following:

- If your organization has well-defined service locations (i.e., premises) AND a premise's service point type controls the types of services offered to the customer, you should probably only use the order transaction to setup new customer information. The new service agreements should be setup using [start / stop](#). See [Prelude To Start/Stop](#) for more information.
- In all other situations, we'd recommend using the [order](#) transaction to start service. If customers typically stop and start service in tandem, we'd recommend the following steps:
 - Stop the customers existing service using [start / stop](#).

- Return to [Control Central - Account Info](#) and select the appropriate campaign from the appropriate portal zone. Keep in mind that the portal zones displayed on this page are based on the user's preferences and security profile. Therefore this zone may not be available to all users.
- When you select the appropriate campaign, you'll be transferred to the [order](#) transaction where you can start the desired package for the customer.

Orders Cannot Be Changed After They Are Completed

The [Lifecycle of an Order](#) indicates that once an order is **complete**, it cannot become pending again. In addition to this rule, we'd like to stress that once an order is **complete**, you cannot make any changes to the order except to add new [log entries](#).

Premise and SP Characteristics May Be Populated From Postal Defaults

If you specify a new premise on the order transaction and you complete the order, a new premise will be created. This premise's characteristics can be populated from two different sources:

- You could have designed the campaign to have its orders capture premise characteristics. You do this by adding miscellaneous fields to the campaign that are premise characteristics.
- You could have set up [postal defaults](#) for the premise's country and postal code.

If a given characteristic is referenced on both of the above, the characteristic value on the campaign is used as the premise's characteristic value.

As described above, service points may be created when a package is selected (it depends on the [SA to Create algorithm](#) that you've used on the selected package). The base package algorithms that create a new service point when a package is selected will populate the service point's characteristics using the information on the postal defaults (if specified). Refer to [STRM-VT](#) for an example of this type of SA to create algorithm.

To Do List Processing

In the base package, there are two To Do lists that may be affected by orders:

- Highlight pending orders. This To Do list contains orders that are in the **pending** state. Refer to [Pending To Do Entries](#) for more information about this To Do list.
- Highlight held orders. This To Do list contains orders that are in the **held** state. Refer to [Held To Do Entries](#) for more information about this To Do list. Refer to [Order - Hold Dialog](#) for more information about holding orders.

Hold and Cancel Reason Codes

In [Lifecycle of an Order](#), we illustrated how a user can **hold** a **pending** order. Before a user can hold an order, they must supply an [Order Hold Reason](#).

The same is true if a user wants to cancel a pending or held order (i.e., they must supply a [Order Cancellation Reason](#)).

These reason codes are purely for audit purposes.

Additional Things Can Happen When An Order Is Completed

An optional plug-in spot exists on [customer class](#) where you can introduce additional logic to be executed when an order is completed for an account that belongs to this customer class.

Another optional plug-in spot exists on [campaign](#) where you can introduce additional logic to be executed when an order is completed for the campaign.

New Premises and Accounts Are Only Created if an Order is Completed

It's important to understand that a premise and account are only created when an order is **completed**. This means you can explore alternatives with a prospect who resides at a new premise without introducing the premise / account. However, the system creates a person if you save an order so that an audit of the customer interaction exists in the system.

Premise Geographic Types Are Populated In An Unusual Way

A [premise's geographic ID's](#) are used by [Control Central](#) when you look for a customer / premise using geographic identifiers.

Geographic coordinates are optional. An [installation option](#) controls whether at least one geographic coordinate is required on every premise.

The following points describe how the geographic ID's are populated on premises that are created when an order is completed:

- If a [postal default](#) exists for the new premise's country / postal code, any geographic identifiers on the postal default are copied onto the new premise.
- If a postal default hasn't been setup for the new premise's country / postal code, the premise will have a single geographic identifier. It is constructed as follows:
 - The geographic type is defaulted from the [campaign](#).
 - The geographic value is set to ***.

Potential error. If neither the campaign nor the postal defaults results in a geographic type AND the [installation record](#) indicates at least one geographic type is required on a premise, an error is generated explaining that either postal defaults or the campaign should be populated with default geographic information.

Account And Premise Usage Are Controlled By The Campaign

When you set up a campaign, you define how the campaign's orders use accounts and premises. Refer to the description of Account Usage and Premise Usage under [Campaign - Main](#) for more information.

Maintaining Orders

The order transaction can be used to satisfy many diverse requirements. For example, you can use an order to:

- Enroll new customers using a single transaction (i.e., you don't have to use the person, premise, service point, and start / stop transactions to enroll a new customer who resides at a new premise).
- Sell new products to existing customers.
- Update existing person, account and premise information using a single transaction.
- Market your services to prospects uploaded from a marketing list (and measure the success of your efforts). If the customer responds to your sales efforts, the system will automatically setup the customer, premise and related service agreements.
- Setup marketing surveys and record your customers' responses.
- Quickly create one-time charges.
- Set up proposals for prospective services (and then send a quotation to the customer for these services).
- And more...

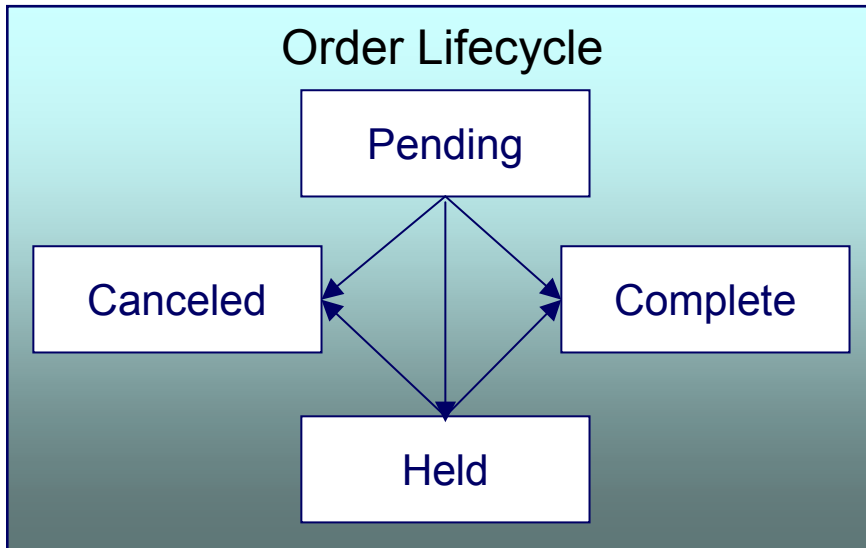
The topics in this section describe the order transaction. Refer to [The Big Picture of Campaigns, Packages and Orders](#) for general information about orders.

Contents

- [Lifecycle of an Order](#)
- [Order User Interface Flow](#)
- [Order - Main](#)
- [Order - Questions & Misc. Fields](#)
- [Order - Log](#)
- [Order - Package Confirmation](#)
- [Order - Hold Dialog](#)
- [Order - Cancel Dialog](#)

Lifecycle of an Order

The following diagram illustrates the lifecycle of an order.

**Pending**

An order starts its life in the **Pending** state. An order remains in this state until you **Cancel**, **Complete** or **Hold** it.

Held

You **Hold** an order when you want to save an order pending future information. For example, you would hold an order if you need to perform research on the perfect rate for the customer before completing the order. A **Held** order can be **Completed** or **Canceled**.

Complete

When a customer selects a package of goods and services, the “V” is updated, service agreements are initiated, and the order becomes **Complete**. No changes may be made to a **Complete** order.

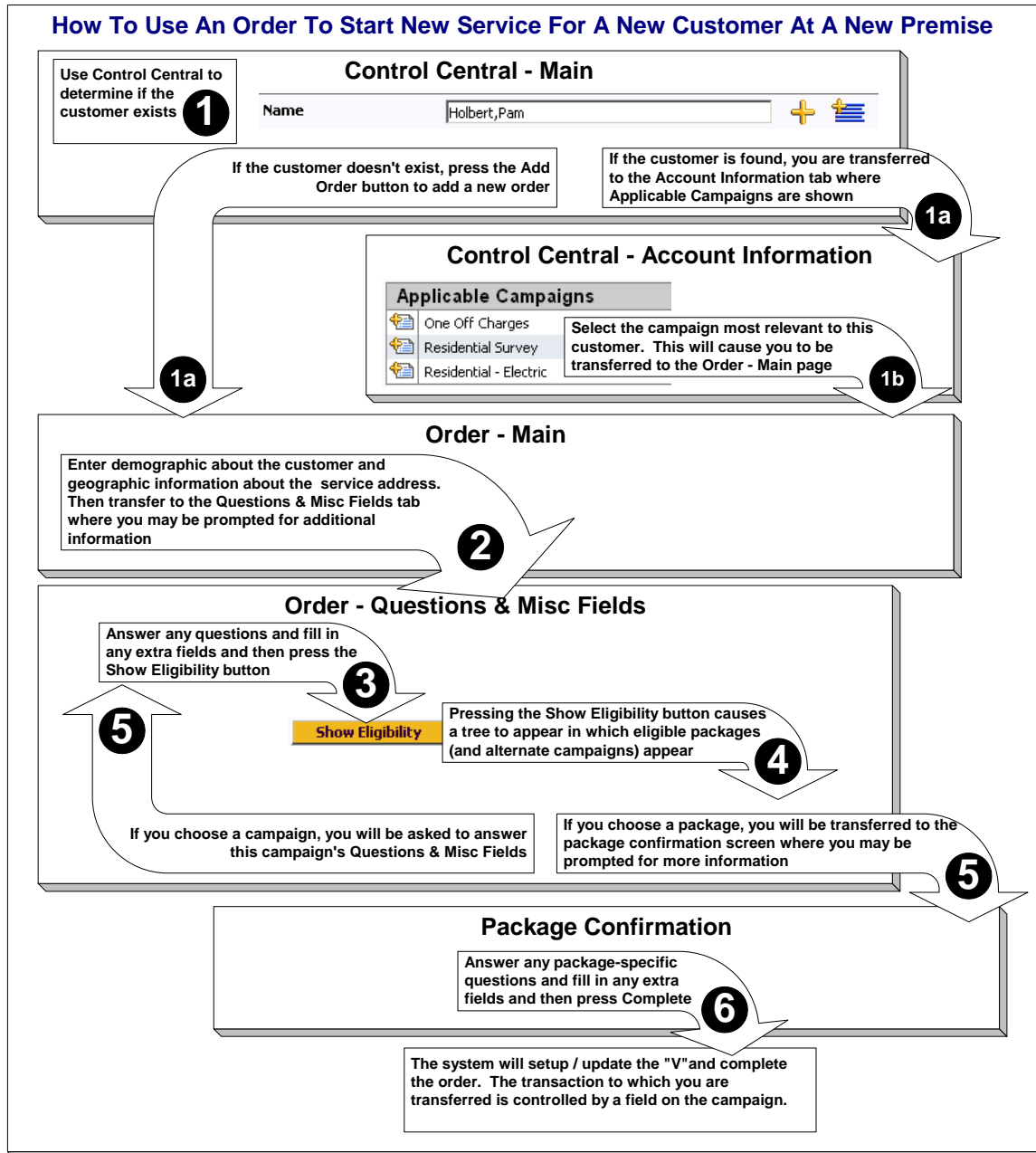
Canceled

A **Pending** or **Held** order may be **Canceled**.

Order User Interface Flow

The diagram that appears below illustrates the anticipated navigation path used to process an order.

Warning! An order's campaign controls many aspects of the user interface flow. The illustration that appears below is a sample of what's possible; it may not mirror how your team has setup your company's campaigns.



Order - Main

This page is used to define demographic and geographic information about the customer / prospect.

Refer to [The Big Picture of Campaigns, Packages and Orders](#) for more information about orders.

Open this page using **Sales & Marketing, Order**.

Refer to [Order User Interface Flow](#) for information about how to use this page.

Description of Page

Order contains a concatenation of the customer's name, campaign, package (if the order is completed), status and start date. **Order ID** is the system-assigned unique identifier of the order. These values only appear after the order is added to the database.

Status contains the state of the order. Refer to [Lifecycle of an Order](#) for information about the valid values and the state transition rules. Additional information may also be displayed based on the status:

- If the order's status is **Held**, the following information is also displayed: the hold reason, the name of the user who held the order, and the date and time when a reminder for the order is due to be created.
- If the order's status is **Canceled**, the following information is also displayed: the cancel reason, the name of the user who canceled the order, and the date and time when the order was canceled.
- If the order's status is **Complete**, the following information is also displayed: the name of the user who completed the order and the date and time when the order was completed.

Press the **Hold** button to hold a **Pending** order. Refer to [Order - Hold Dialog](#) for a description of the pop-up window used to hold an order.

Press the **Cancel** button to cancel a **Pending** or **Held** order. [Refer to Order - Cancel Dialog](#) for a description of the pop-up window used to cancel an order.

If the **Campaign** has been setup to allow the completion of orders without selecting a package, a button appears to the right of **Cancel**. The label that appears in the button is defined on the **Campaign**. When pressed, the order is completed. When the order is completed,

- [The "V"](#) is setup / updated.
- All questions and miscellaneous fields (defined on the next tab) are posted to the appropriate object.
- [Optional plug-ins](#) are executed.
- You will be transferred to the order completion transaction that's defined on the **Campaign** (if any).

If the order references an existing person, account, or premise and the related person / account / premise has changed after their information was snapshot on the order, the **Out of Sync** section appears. This section highlights which objects have changed since the order was last saved. In addition, the **Sync** button appears. When pressed, the order is updated with the current field values for these objects.

An order's **Campaign** controls many aspects of the order's behavior. For example, it controls:

- The type of information defaulted onto an order.
- The eligibility-oriented questions that are displayed on the next tab.
- The superset of packages that can be offered to a customer.
- In addition to the above, campaigns also control high-level [eligibility rules](#) and [business process flows](#).

An order's **Campaign** is defaulted as follows:

- When you select a customer on [Control Central - Main](#), you are automatically transferred to the [Account Information](#) portal. One of the [zones](#) on this portal page contains the campaigns that can be offered to the customer. If you select a campaign from this zone, you are transferred to the [Order](#) transaction with the respective Campaign defaulted. Note, the **Applicable Campaigns** zone only appears if the CSR has modified their preferences to display this zone.
- In all other situations, the order's campaign defaults from the [Installation Record](#).

You may change an order's **Campaign** at will. Note that changing the campaign will result in the **Questions & Misc Fields** (the next tab) being refreshed with values associated with the new campaign. However, other values typically defaulted from the campaign (such as default customer class) do not get refreshed so that data you may have entered does not get overwritten.

Start Date defines the date on which service will start. This date becomes the start date on service agreements created when this order is completed. If you need to change the start date on these service agreements, transfer to [Start/Stop](#) (where all of the **pending start** service agreements will appear).

The topics that follow describe the remaining sections on this page

Contents

- [Order - Main - Person Information](#)
- [Order - Main - Service Address](#)
- [Order - Main - Account Information](#)

Order - Main - Person Information

The **Person Information** section contains demographic information about the customer / prospect.

Warning! This page contains a subset of person-oriented information. If you need to define other person-oriented information, transfer to [Person - Main](#) after the order is completed.

If this is a new customer, select **Create New Person** and enter demographic information in the fields that follow. Otherwise, select **Use Existing Person** and use the **Name** field or press the search button to find the customer. After you select an existing person, the following events takes place:

- The person's name appears beneath the person search (and the person context menu can now be used to drill down to person-oriented pages). Note, the person's name is formatted by a plug-in algorithm on the [installation record](#). Refer to the base package's [name format algorithm](#) for an example. If you prefer different formatting logic, your system administrator should configure the system appropriately.
- If any of the Questions & Misc Fields (on the next tab) are used to capture person-oriented characteristics, the person's existing characteristic values are defaulted.
- If the selected person has at least one account, information about the account is displayed in the **Account Information** section (if the **Campaign** makes uses of accounts). In addition, if any of the Questions & Misc Fields (the next tab) are used to capture account-oriented characteristics, the account's existing characteristic values are populated in the respective rows on this tab.
- Information about the person is displayed in the fields that follow.

Use **Person/Business** to indicate if the customer is a **Person** or a **Business**. This value controls how the person's primary name is validated. Note, for new customers, this value defaults based on the **Campaign**.

Names are used by [Control Central](#) when you look for a customer based on their name. In addition, a person's primary name is the addressee on the person's bill unless overridden by the **Override Mailing Name** (defined on [Person - Correspondence Information](#)). To modify a name, simply move to a field and change its value. To add a new name, press + to insert a row, then fill in the information for each field. The following fields display:

- Use **Name Type** to indicate if the name is an **Alias**, **Alternate Representation**, **Doing Business As**, **Legal**, or **Primary** name. Note, for new persons, a value of **Primary** is defaulted.
- Use **Person Name** to define the person's name. Note well, the name is case sensitive.

Alternate representations of a person's name. You would use an **Alternate Representation** for a person's name when you have an alternate ways to define the person's primary name. Alternate representations are typically used in countries that use multiple character sets (e.g., the **Primary** name is entered in Chinese, the **Alternate Representation** is entered in English). When a person has an alternate name, both the main and alternate names can be used to search for a person. The **Alternate Representation Name Type** only appears if you have enabled alternate addresses on the installation record. Refer to the description of the Alternate Representation field under [Installation - Main](#) for more information.

Validation is performed by a plug-in. The validation that is applied to **Person Name** (e.g., a comma separating the last and first name – **Smith,Patricia**) is controlled by a plug-in algorithm on the [installation record](#). Refer to the base package's [name validation algorithm](#) for an example. If you prefer different validation logic, your system administrator should configure the system appropriately.

Phone numbers are used by [Control Central](#) to look for accounts and persons. For new customers, the phone type(s) default based on the **Campaign**. To modify a telephone number, simply move to a field and change its value. To add a new telephone number, press + to insert a row, then fill in the information for each field. The following fields display:

- **Phone Type** indicates the type of phone number, e.g., Home, Mobile, Business, ...
- Use **Phone Number** to define the telephone number. Enter the telephone number in the format described by the **Phone Format**.

Formatting is performed by a plug-in. The format that is applied to a **Phone Number** is controlled by the algorithm that is plugged in on the respective **Phone Type**. You can also have this plug-in convert input values into formatted values, e.g., this algorithm could transform all numbers into a formatted string. If you prefer a different format, your system administrator should configure this algorithm appropriately.

- Enter the **Extension**, if any, of the telephone number.

A person's **ID** has several uses:

- It is used by [Control Central](#) when you look for a customer / premise based on their ID.

- It is used to highlight potential duplicate persons.
- [Control Central](#) displays a person's primary identification in the search results area to help a user identify the customer when multiple customers match the search criteria.

The person ID usage flag on the [installation record](#) indicates whether or not at least one id for a person is **required** or **optional**. The ID Type defaults from the [installation record](#) based on the **Person Type** (**Person** versus **Business**). The following fields are used to define a customer's primary ID.

- Indicate the type of identification in the drop down.
- Enter the identification number in the adjacent fields. Please note that if the ID number should be formatted (e.g., dashes in an American social security number), you do not have to enter the dashes. Rather, you can enter the information as a contiguous value and the system will format this for you. The format is shown in the adjacent **Identifier Format** column.

Formatting is performed by a plug-in. The format that is applied to a person ID number (e.g., dashes in an American social security number) is controlled by the algorithm that is plugged in on the respective [ID Type](#). If you prefer a different format, your system administrator should configure this algorithm appropriately.

A person can have many forms of identification. The order transaction only caters for a single form of ID (as this is the most typical situation). If you want to specify additional person ID's, you must transfer to [Person - Main](#) after the person is created (persons are created when the order is saved).

Specify the customer's **Email ID** (if any).

Define the **Language** in which the person prefers their bills and correspondence printed.

Default note. The person's language defaults from [Installation Options - Person](#).

Refer to [Customer Language](#) for more information on options for supporting multiple languages for your customers.

Use **Life Support / Sensitive Load** to indicate if the person has life support or sensitive load equipment. Valid values are: **LS/SL** (i.e., the person has life support / sensitive load equipment), **None**. If the customer has **Life Support / Sensitive Load** equipment, indicate such in the adjacent comment field.

A premise can also have life support / sensitive load information. If the equipment is physically linked to the premise (e.g., a hospital has life support equipment), you should NOT specify the life support information on the person. Rather, after the premise is added, transfer to [Premise - Misc](#) and specify life support information on the premise.

Life support affects alerts and C&C. If life support / sensitive load information is specified, an alert will appear when the person is displayed on control central. In addition, if a person has life support or sensitive load equipment, it is possible for a different credit & collection severance process to be kicked off if the person's account associated has overdue debt. Refer to [Designing Your Severance Procedures](#) for more information.

Order - Main - Service Address

The **Service Address** section contains geographic information about the service address:

Warning! This page contains a subset of premise-oriented information. If you need to define other premise-oriented information, transfer to [Premise - Main](#) after the order is completed.

Whether or not a premise is required or even allowed on an order is controlled by an option on the **Campaign**. If the **Campaign** disallows the use of a premise, this section will be suppressed. If the **Campaign** allows the use of a premise, an indication of whether a premise is optional or required is displayed adjacent to **Service Address**.

If a premise is allowed on the order, use the dropdown at the top of this section to tell the system about this order's premise:

- Select **No Premise Information** if you do not want to associate a premise with this order (e.g., for a one-time charge).
- Select **Create New Premise** if this is a new premise. You should enter the new premise's address in the remaining fields in this section.
- Select **Use Existing Premise** if an existing premise should be used on the order. Use the **Address** and **City** fields or press the search button to find the premise. Refer to [Control Central - Search Facilities](#) for suggestions in respect of how to use these fields. After you select an existing premise, the following events takes place:
 - The premise's description appears beneath the premise search fields (and the premise context menu can now be used to drill down to premise-oriented pages).
 - If any of the Questions & Misc Fields (the next tab) are used to capture premise-oriented characteristics, the premise's existing characteristic values are defaulted.
 - Information about the premise is displayed in the fields that follow.

Use **Premise Type** to describe the type of service location.

The address's constituent fields vary based on the **Country**. Please refer to the [Country](#) page for more information.

Populate **Trend Area** if the **Postal** is blank. The **Postal** and the **Trend Area** cannot both be blank as the **Trend Area** is a required field on the premise and its value is typically defaulted from [postal defaults](#).

Default note. The values of several fields on this and the next page default based on the **Country** and the **Postal Code**. Refer to [Setting Up Premise & Service Point Postal Defaults](#) for more information. Also note, there are several premise-oriented fields that do not appear on this transaction that will be populated on the premise based on the **Country** and **Postal Code** (e.g., characteristics, time zone, trend area, geo ID's).

Use **CIS Division** to define the jurisdiction in which the premise is located.

Order - Main - Account Information

The **Account Information** section contains account-oriented information.

Warning! This page contains a subset of account-oriented information. If you need to define other account-oriented information, transfer to [Account - Main](#) after the order is completed.

Whether or not an account is required or even allowed on an order is controlled by an option on the **Campaign**. If the **Campaign** disallows the use of an account, this section will be suppressed. If the **Campaign** allows the use of an account, an indication of whether an account is optional or required is displayed adjacent to **Account Information**.

If an account is allowed on the order, use the dropdown at the top of this section to tell the system about this order's account:

- Select **No Account Information** if you do not want to associate an account with this order (e.g., for a person-oriented marketing survey).
- Select **Create New Account** if this is a new account. You should enter the new account's information in the remaining fields in this section.
- Select **Use Existing Account** if an existing account should be used on the order. If an existing person was selected above, one of the person's accounts (if any) will be automatically displayed in the **Account Information**. In addition, if the order references an existing person, the account search will be limited to accounts associated with this person. If an **Existing Account** is selected, the following takes place:
 - Basic information about the account appears beneath the search area (and the account context menu can now be used to drill down to account-oriented pages).
 - If any of the Questions & Misc Fields on the next tab are used to capture account-oriented characteristics, the account's existing characteristic values are defaulted.
 - Information about the account is displayed in the fields that follow.

Customer Class plays a part in:

- If and when a customer is subject to late payment charges.
- The account's default collection class and when the account debt monitor reviews an account. Refer to [The Big Picture of Credit & Collections](#) for more information about how and when an account's debt is reviewed.
- And several other functions. Refer to and [Setting Up Customer Classes](#) for more information.

Default note. For new accounts, **Customer Class** defaults based on the **Campaign**.

The optional Account **Management Group** controls the roles assigned to To Do entries associated with an account. Refer to [Setting Up Account Management Groups](#) for more information.

Default note. For new accounts, **Management Group** defaults based on the **Campaign**.

The remaining fields are used to describe how the bill should be routed to the customer.

Warning! When an order is completed, several fields that do not appear on this page that reside on [Account - Person](#) are populated. For example, **Account Relationship Type**, **Bill Format**, **Number of Copies**, **Receives Notification** are all populated with default values. If you need to override the default values, transfer to [Account - Person](#) after the account is added.

Use **Bill Route Type** to define how the bill is sent to the customer. This field's value defaults from the [Installation Record](#). This value is important as it controls many options on the remainder of this page. If the **Bill Route Type** indicates that bills are routed via Fax, the person's fax number is displayed adjacent (the system knows which of a [person's phone numbers](#) is a fax number by the **Phone Type**). If the **Bill Route Type** indicates that bills are routed via Email, the person's **Email** address is displayed adjacent. If the **Bill Route Type** indicates that bills are routed via the Postal service, you must choose an appropriate **Bill Address Source** to define which address should be used. Refer to [Setting Up Bill Route Types](#) for more information about bill route types.

If the **Bill Route Type** indicates that bills are routed via the Postal service, you must choose the appropriate **Bill Address Source** to define which address should be used. This field will be protected for other **Bill Route Types**.

- Choose **Mailing Premise on Account** if bills should be sent to the address associated with the Mailing Premise on the first page. This address is displayed adjacent.
- Choose **Person** if bills should be sent to the [person's mailing address](#). This address is displayed adjacent.
- Choose **Account Override** if bills should be sent to an override address specified below. Typically, you would only choose this option if the person has multiple accounts and each account's bills should be sent to a different address.

If you select a **Bill Address Source** of **Account Override**, you must enter the address to which bills will be sent in the following address constituents. The number and type of address constituents is based on the **Country** (refer to [Defining Countries](#) for more information on address constituents). These fields will be protected for other **Bill Address Sources**. Note, the **Country** defaults from [Installation Options - System](#) and several other constituents default based on the **Country** and **Postal Code** if a [Postal Default](#) exists for the postal code.

Order - Questions & Misc. Fields

This page has several purposes:

- The order's Campaign can define questions to be posed to customers when an order is taken. For example, you could ask the customer "marketing survey" questions (e.g., please rate our service, how do you heat your home?, etc.). This page allows you to capture the customer's response.
- The order's Campaign can also be setup to capture additional fields to be updated on the order's person / account / premise when the order is completed. For example, you could indicate that the account's override due date and automatic payment options should be captured when an order is taken. This page allows you to capture these field values.
- The order's Campaign can also be setup to pose eligibility-oriented questions that qualify the customer for packages of goods and services. For example, you might have a campaign with packages that can only be selected if the customer commits to a year's worth of service. This page allows you capture the customer's response to these types of questions.

- After entering the above information (and the information on the Main page), you can use this page to select a package of goods and services for the customer.
- You can use this page to switch the campaign on the order (which would result in new questions and miscellaneous fields appearing).

Refer to [The Big Picture of Campaigns, Packages and Orders](#) for more information.

Open this page using **Sales & Marketing, Order** and then navigate to the **Questions & Misc Fields** tab.

Refer to [Order User Interface Flow](#) for information about how to use this page.

Description of Page

Order contains a concatenation of the customer's name, campaign, package (if the order is completed), status and start date. **Order ID** is the system-assigned unique identifier of the order. These values only appear after the order is added to the database.

Status contains the state of the order. Refer to [Lifecycle of an Order](#) for information about the valid values and the state transition rules. Additional information may also be displayed based on the status:

- If the order's status is **Held**, the following information is also displayed: the hold reason, the name of the user who held the order, and the date and time when a reminder for the order is due to be created.
- If the order's status is **Canceled**, the following information is also displayed: the cancel reason, the name of the user who canceled the order, and the date and time when the order was canceled.
- If the order's status is **Complete**, the following information is also displayed: the name of the user who completed the order and the date and time when the order was completed.

Press the **Hold** button to hold a **Pending** order. Refer to [Order - Hold Dialog](#) for a description of the pop-up window used to hold an order.

Press the **Cancel** button to cancel a **Pending** or **Held** order. [Refer to Order - Cancel Dialog](#) for a description of the pop-up window used to cancel an order.

If the **Campaign** has been setup to allow the completion of orders without selecting a package, a button appears to the right of **Cancel**. The label that appears in the button is defined on the **Campaign**. When pressed, the order is completed. When the order is completed,

- [The "V"](#) is setup / updated.
- All questions and miscellaneous fields (defined on the next tab) are posted to the appropriate object.
- [Optional plug-ins](#) are executed.
- You will be transferred to the order completion transaction that's defined on the **Campaign** (if any).

If the order references an existing person, account, or premise and the related person / account / premise has changed after their information was snapshot on the order, the **Out of Sync** section appears. This section highlights which objects have changed since the order was last saved. In addition, the **Sync** button appears. When pressed, the order is updated with the current field values for these objects.

The number and type of **Questions & Misc Fields** is controlled by the [Campaign](#).

Some questions can be suppressed. The campaign contains an indication if each question is account-oriented, person-oriented, premise-oriented or order-oriented. If the question is account-oriented and no account is defined on the **Main** tab, the question will be suppressed. If the question is premise-oriented and no premise is defined on the **Main** tab, the question will be suppressed.

Each row in this grid has a **Prompt** and a **Response**. The value of each **Prompt** is defined on the **Campaign**. The **Response** holds the answer supplied by the customer. Please note the following about the **Response** field:

- On a **Campaign**, you can indicate which questions / misc. fields are required. The Response for a required field is prefixed with an asterisk - *.
- The value of a Response can default from a variety of sources:
 - If an existing person was selected on the **Main** page AND the response is used to capture a [person characteristic](#), the existing characteristic value will be defaulted.
 - If an existing premise was selected on the **Main** page AND the response is used to capture a [premise characteristic](#), the existing characteristic value will be defaulted. If no value exists, the characteristic values will default based on the postal defaults associated with the service address's **Country** and **Postal Code**.
 - If an existing account was selected on the **Main** page AND the response is used to capture an [account characteristic](#), the existing characteristic value will be defaulted.
 - Otherwise, the characteristic value is defaulted from the **Campaign**.
- The type of field used to capture a Response is controlled by the Campaign. Depending on the type of Response you may see [Go To](#) buttons and [Search](#) buttons.

After entering the **Responses** to the **Questions & Misc Fields**, press the **Show Eligibility** button.

The following may occur when this button is pressed:

- If a new customer is specified on the Main page, a warning will advise you that a new person will be added. The system needs to add a new person so that an audit of the customer interaction exists.
- The tree expands to show the Order - Eligibility Tree containing package, ineligible package, error package, and other campaign nodes:
 - Each **Package** node contains a package that is linked to the campaign whose [Eligibility Criteria](#) allows its selection by the customer. You can expand this node to see the various service agreements that will be created if you select the package. Selecting one of these packages causes the [Order - Package Confirmation](#) page to appear in which you can confirm your selection.

- The **Ineligible Package(s)** node contains packages whose [Eligibility Criteria](#) prevents their selection by the customer.
- The **Error Packages** node contains packages whose [Eligibility Criteria](#) have returned errors (if any).
- The **other campaign(s)** node contains alternative campaigns that may be substituted for the current campaign on the package. Selecting one of these campaigns will cause the campaign to be changed on this order. Changing a campaign on an order will result in new questions and miscellaneous fields appearing. Refer to [The Big Picture Of Campaign Eligibility](#) for more information.

Some nodes can be suppressed. To increase page performance with large campaigns, you can define a [feature configuration](#) to suppress the display of the Ineligible Packages, Error Packages or Other Campaign nodes. If a node is suppressed, it does not appear on the eligibility tree and the system does process the information necessary to display the node.

Order - Log

This page contains an audit trail of significant events that transpired in the history of this order.

Open this page using **Sales & Marketing, Order** and then navigate to the **Log** tab.

Description of Page

Order contains a concatenation of the customer's name, campaign, package (if the order is completed), status and start date. **Order ID** is the system-assigned unique identifier of the order. These values only appear after the order is added to the database.

Status contains the state of the order. Refer to [Lifecycle of an Order](#) for information about the valid values and the state transition rules. Additional information may also be displayed based on the status:

- If the order's status is **Held**, the following information is also displayed: the hold reason, the name of the user who held the order, and the date and time when a reminder for the order is due to be created.
- If the order's status is **Canceled**, the following information is also displayed: the cancel reason, the name of the user who canceled the order, and the date and time when the order was canceled.
- If the order's status is **Complete**, the following information is also displayed: the name of the user who completed the order and the date and time when the order was completed.

Press the **Hold** button to hold a **Pending** order. Refer to [Order - Hold Dialog](#) for a description of the pop-up window used to hold an order.

Press the **Cancel** button to cancel a **Pending** or **Held** order. [Refer to Order - Cancel Dialog](#) for a description of the pop-up window used to cancel an order.

Campaign contains basic information about the order's campaign.

Package only appears if the order has been completed. It contains basic information about the package.

Start Date contains the date on which the service agreement's created by the order will be started.

The **Log** contains an audit trail of significant events that transpired in the history of this order. There are two types of entries in the log:

- The system automatically adds log entries when key events occur. These entries may not be removed or changed.
- Users may add log entries by pressing the + button and then specifying the Details.

Order - Package Confirmation

The Package Confirmation page allows you to check the package you selected before [the V](#) is created / updated for the customer. This page opens when you select a package from the eligibility tree on [Order - Questions & Misc Fields](#).

Warning! If you press the **Complete** button, a variety of objects may be created (service agreements, service points, a premise, an account, field activities, etc.). If you need to change any of these objects after order completion, you must transfer to the respective page on which the object is maintained. You can also use [Start / Stop](#) to make changes to the service agreements created by this transaction.

Description of Page

Order contains a concatenation of basic information about the order. **Order ID** is the system-assigned unique identifier of the order. These values only appear after the order is added to the database.

Campaign contains basic information about the order's campaign.

Package contains basic information about the package.

Premise is the address at which service will be delivered.

Person is the name of the customer.

Account is the name of the customer and their customer class.

Start Date contains the date on which the service agreement's created by the order will be started.

The **Questions & Misc Fields** grid contains the questions and miscellaneous fields that appeared on the order. Note well, it is possible to design a campaign and a package where some fields are optional on the campaign but required for a specific package. The **Response** of all required fields for the selected package are prefixed with an asterisk. Refer to [Order - Questions & Misc Fields](#) for information about the rows in this grid.

Changing your answers could cause the package to become ineligible. As described earlier, the fields you enter on an order play are what the system uses to determine if a package can be selected for a customer. If you change the answers to the **Questions & Misc Fields**, the package may no longer be eligible and an error will be presented to you. You will have the choice of reverting to the original answers or returning to the order page where you can select an eligible package.

Additional questions / misc fields are possible. It is possible to design a campaign where certain field values are only visible on specific packages. In this scenario, the **Questions & Misc Fields** grid may contain additional rows for a given package. Refer to [Designing Campaigns and Packages](#) for more information.

The **Instructions** grid contains package-specific suggestions. These instructions are defined on the [package](#).

The **Messages** grid contains system-generated messages that summarize what will happen if you press the **Complete** button.

Press the **Complete Order** button if you are satisfied with the information on the order. Otherwise, press **Do Not Use this Package** to change or cancel the order.

Order - Hold Dialog

The **Order - Hold** dialog opens if you press the **Hold** button on the order transaction. You use it to define why the order is being held and when you want to be reminded to check up on the order.

Description of Page

Use **Hold Reason** to define why the order is being held.

The following fields are used to control the [To Do entry](#) that will be created to remind you about this order.

- Use **Reminder Date / Time** to define the latest date on which the To Do entry should be created. The reason we indicated this should be the latest date is because the background process (known by the batch control ID of **TD-ECBK**) that's responsible for creating these To Do entries has a parameter called "lead time". This parameter is used to define the number of days before the **Reminder Date** that the To Do entry should be created.
- If the To Do entry should be addressed to a group of users, choose a **Reminder Type** of **Send to Role** and enter the user group's **To Do Role**. Note, the To Do Role will default from the order's campaign.
- If the To Do entry should be addressed to a specific user, choose a **Reminder Type** of **Send to User** and enter the user's User ID in **To Do User**.
- **Log Notes** are used to describe why the order is being held. This information appears in the [log entry](#) that is created by the system when you hold an order.

If there are any To Do entries (with a status of **Open** or **Being Worked On**) that drill down to this order, a message is displayed indicating that existing To Do entries for this order will be completed. If the order is already in **Hold** status, existing To Do entries are completed when you click the OK button and a new To Do entry is created when the background process responsible for creating the To Do entries runs.

Order - Cancel Dialog

The Order - Cancel dialog opens if you press the **Cancel** button on the order transaction. You use it to define why the order is being canceled.

Description of Page

Use **Cancel Reason** to define why the order is being canceled.

Use **Log Notes** to describe why the order is being canceled. This information appears in the [log entry](#) that is created by the system when you cancel an order.

Maintaining Campaigns

When you add an [order](#), you must reference its [campaign](#). An order's campaign controls:

- The type of information defaulted onto an order. For example, you can setup a campaign to default a given account management group on all orders linked to this campaign. This account management group will then default onto accounts created when the campaign's orders are completed.
- The eligibility-oriented questions that are posed to the customer when an order is taken. For example, the questions indicated above – Who is your current energy service provider? / Would you like to pay automatically? / What is your date of birth? – are all defined on the order's campaign.
- The superset of packages that can be offered to a customer whose order references this campaign. An order's campaign defines the types of packages that may be selected.
- In addition to the above, campaigns also control high-level [eligibility rules](#) and [business process flows](#).

The topics in this section describe the campaign transaction.

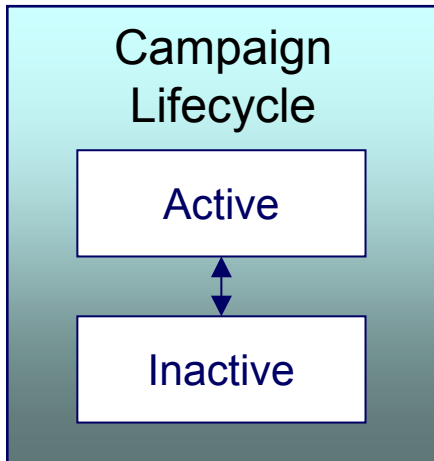
Refer to [The Big Picture of Campaigns, Packages and Orders](#) for general information about campaigns. Refer to [The Big Picture Of Campaign Eligibility Rules](#) for a discussion of how to control which campaigns can be used for different types of customers. Refer to [Designing Campaigns and Packages](#) for guidelines describing how to design your campaigns.

Contents

- [Lifecycle of a Campaign](#)
- [Campaign - Main](#)
- [Campaign - Questions and Misc. Fields](#)
- [Campaign - Algorithm](#)
- [Campaign - Package](#)

Lifecycle of a Campaign

The following diagram illustrates the lifecycle of a campaign.

**Active**

An **Active** campaign may have orders created in respect of it.

Inactive

By **Inactivating** the campaign, you are preventing it from being used on future orders. In addition, **Inactive** campaigns do not appear in the [Applicable Campaigns](#) zone.

Campaign - Main

This page is used to define basic information about a campaign.

Refer to [The Big Picture of Campaigns, Packages and Orders](#) and [Designing Campaigns and Packages](#) for more information.

Open this page using **Sales & Marketing, Campaign**.

Description of Page

Campaign is the user-defined code that uniquely identifies the campaign. The campaign's description appears adjacent.

Description briefly describes the campaign.

Campaign Status describes the state of the campaign. Refer to [Lifecycle of a Campaign](#) for more information.

The following fields contain values to be defaulted onto this campaign's orders. The user who takes the order may override these default values. Note well, these default values are only applicable for new persons – they will not override field values defined on existing persons.

- The system requires all persons to be defined as either a **Person** or a **Business**. Use **Default Person / Business** to define the default value for persons created by this campaign's orders.
- Every phone number defined for a person must reference a [phone type](#). Use **Default Phone Type 1** and **Default Phone Type 2** to define the first two phone types defaults for persons created by this campaign's orders.

Premise Usage controls if and how this campaign's orders use premises. You have the following options:

- Select **Not Allowed** if this campaign's orders do not reference a premise.
- Select **Optional** if this campaign's orders do not have to reference a premise. This option also means that a new OR existing premise may be specified on the order. Note, if an order references a new premise, the premise will be created when the order is completed.
- Select **Optional - Existing Premise Only** if this campaign's orders do not have to reference a premise. This option also means that only an existing premise may be specified on an order.
- Select **Required** if this campaign's orders must reference a premise. This option also means that a new OR existing premise may be specified on the order. Note, if an order references a new premise, the premise will be created when the order is completed.
- Select **Required - Existing Premise Only** if this campaign's orders must reference a premise. This option also means that only an existing premise may be specified on an order.

The following fields contain values to be defaulted onto this campaign's orders. The user who takes the order may override these default values. Note well, these default values are only applicable for new premises – they will not override field values defined on existing premises.

- The system requires all premises to reference a [premise type](#). Use **Default Premise Type** to define the default value for premises created by this campaign's orders.
- Every geographic ID defined for a premise must reference a [geographic type](#). Use **Default Premise Geo Type** to define the default value for premises created by this campaign's orders. Refer to [Premise Geographic Types Are Populated In An Unusual Way](#).

Account Usage controls if and how this campaign's orders use accounts. You have the following options:

- Select **Not Allowed** if this campaign's orders do not reference an account.
- Select **Optional** if this campaign's orders do not have to reference an account. This option also means that a new OR existing account may be specified on the order. Note, if an order references a new account, the account will be created when the order is completed.
- Select **Optional - Existing Account Only** if this campaign's orders do not have to reference an account. This option also means that only an existing account may be specified on an order.
- Select **Required** if this campaign's orders must reference an account. This option also means that a new OR existing account may be specified on the order. Note, if an order references a new account, the account will be created when the order is completed.
- Select **Required - Existing Account Only** if this campaign's orders must reference an account. This option also means that only an existing account may be specified on an order.

The following fields contain values to be defaulted onto this campaign's orders. The user who takes the order may override these default values. Note well, these default values are only applicable for new accounts – they will not override field values defined on existing accounts.

- The system requires all accounts to belong to a [customer class](#). Use **Default Customer Class** to define the default value for accounts created by this campaign's orders.
- An account may reference an [account management group](#). Use **Default Acct Management Group** to define the default value for accounts created by this campaign's orders.

If a user [holds](#) one of the campaign's orders, they can have a [To Do entry](#) generated on a future date to remind them about the order. Use **Default Hold To Do Role** to define the [group of users](#) to whom the To Do entry will be sent. Note well, the user who holds an order can indicate that the related To Do entry should be addressed to a specific user ID rather than a **Role**.

Use **Order Open Option** to define the tab page that is displayed when a user adds a new order for this campaign. This field is only relevant when the order transaction is invoked in "add mode" and a campaign is passed to this transaction (e.g., if a user adds an order from the [Applicable Campaigns Zone](#)).

A user can be automatically transferred to a different transaction when they complete an order. Refer to [Supported Business Processes](#) for examples of transactions that we anticipate being used. Use **Navigation Option** to define this transaction. Note,

- Leave **Navigation Option** blank if the user should remain on the order transaction after completion.
- You can setup **Navigation Options** to transfer to the user to a tab other than the main tab.

If a user is allowed to complete this campaign's order's without selecting a package,

- Turn on **Allow Order Compl w/o Package** and
- Use **Order Compl w/o Package Label** to define the message to appear in the button used to complete such orders.

Refer to [Campaigns Without Packages](#) for examples of campaigns where you would enable the above fields.

Use **Long Description** to provide an overview of this campaign.

The grid at the bottom of the page summarizes this campaign's questions and miscellaneous fields. To change this information, press the adjacent go to button (which will transfer you to the [Questions and Misc Fields](#) tab). To add additional information, transfer to the [Questions and Misc Fields](#) tab and press the + button.

Campaign - Questions and Misc. Fields

This page is used to define the following type of information:

- An order's campaign can define questions to be posed to customers when an order is taken. For example, you could ask the customer "marketing survey" questions (e.g., please rate our service, how do you heat your home?, etc.).
- An order's campaign can also be setup to capture additional fields to be updated on the order's person / account / premise when the order is completed. For example, you could indicate that the account's override due date and automatic payment options should be captured when an order is taken.
- An order's campaign can also be setup to pose eligibility-oriented questions that qualify the customer for packages of goods and services. For example, you might have a campaign with packages that can only be selected if the customer commits to a year's worth of service.

Refer to [Designing Campaigns and Packages](#) for information describing how to design this type of information for your campaigns.

Open this page using **Sales & Marketing, Campaign** and navigate to the **Question & Misc Fields** tab.

Description of Page

Warning! This information is not intuitive; we strongly recommend that you follow the guidelines under [Designing Campaigns and Packages](#) before attempting to enter this information.

Campaign is the user-defined code that uniquely identifies the campaign. The campaign's description appears adjacent.

The **Questions & Misc Fields** scroll contains one row for each question / miscellaneous field captured on this campaign's orders. The following fields are used:

- **Prompt on Order** contains the prompt that is shown to the user when the [order](#) is taken. Refer to [Determine The Properties Of Every Miscellaneous Field](#) for examples.
- **Brief Description** is simply a brief description of the question / miscellaneous field.
- **Sort Sequence** controls the order in which the question appears on the order.
- **Applicability** controls whether the field is **Optional**, **Required**, or **Only Applicable on Package**.
 - A value of **Required** means the order taker must enter a valid value before packages are displayed for selection.
 - A value of **Optional** means the order taken can leave the field blank prior to package selection. You can make an **Optional** field **Required** on individual packages.
 - A value of **Only Applicable on Package** is used for fields that are only captured for specific packages. Fields of this type are suppressed prior to package selection. When you setup a [package](#), you indicate if the field is applicable (if so, the user is only prompted for the field if the package is selected).
- **Type of Response** controls where the field is stored when the order is completed and how it is validated. You have two options: **Characteristic** or **Column**.
 - If a field is a **Characteristic**, you must define which entity the characteristic value is to be stored on when the order is completed (**Person**, **Account**, **Premise** or **Order**). You must also define the **Characteristic Type** as this controls the characteristic value entered by the user.
 - If a field's value resides in a **Column** (as opposed to a characteristic value), you must define the [column reference](#). A field's **Column Reference** controls three things:
 - How the field's value is retrieved. A field's value is retrieved when an existing person / account / premise is referenced on an order.
 - How the field's value is validated. A field's value is validated before package eligibility is determined and before the order is completed.
 - How the field's value is posted (i.e., updated on the database). A field's value is posted when an order is completed.

Refer to [Extract Column References From Miscellaneous Fields](#) for more information about column references.

- If a field's value resides in a **Column** (as opposed to a characteristic value), you must also define whether the question / field is dependent on the order referencing an account or premise. You do this by populating **Question Dependency** with one of the following values:
 - Select **Must Have Account** if this question / field should only appear if the order references an account (new OR existing).
 - Select **Must Have Premise** if this question / field should only appear if the order references a premise (new OR existing).
 - Select **No Dependency** if this question / field should always appear.

Refer to [Determine Account / Premise Dependencies For Column References](#) for more information.

- **Default Column Value / Default Char Value** controls whether the order taker is supplied with a default value when an order is created. Note well: for **Optional** and **Only Applicable on Package** fields, you can specify a different default value on each package.

Campaign - Algorithm

This page defines plug-in algorithms that are executed for orders associated with this campaign. Open this page using **Sales & Marketing, Campaign** and navigate to the **Algorithm** tab.

Description of Page

Campaign is the user-defined code that uniquely identifies the campaign. The campaign's description appears adjacent.

The grid contains **Algorithms** that control campaign-oriented functions. You must define the following for each plug-in algorithm:

- Specify the **System Event** with which the algorithm is associated (see the table that follows for a description of all possible events).
- Specify the **Sequence** and **Algorithm** for each **System Event**. You can set the **Sequence Number** to 10 unless you have a **System Event** that has multiple **Algorithms**. In this case, you need to tell the system the **Sequence** in which they should execute.

You can define algorithms for the following **System Events**:

System Event	Optional / Required	Description
<i>Campaign Eligibility</i>	Required	<p>A campaign's eligibility algorithms control two things:</p> <ul style="list-style-type: none"> - Whether the campaign appears in the Applicable Campaigns Zone. This content zone suggests possible campaigns for the current customer displayed on control central. You might want to use this zone if you have different campaigns that are offered to different types of customers. Refer to Real-time Marketing of Additional Services to a Customer for an example of how this content zone can be used to "up sell" additional services. - Whether the campaign appears on an order's eligibility tree. Campaigns that appear in this tree are alternate campaigns that may be used on an order. Refer to Real-time Marketing of Services to a Prospect for an example of how

		<p>an order's eligibility tree can contain alternate campaigns for an order.</p> <p>Refer to The Big Picture Of Campaign Eligibility for information about these algorithms.</p> <p>Click here to see the algorithm types available for this system event.</p>
<i>Order Completion</i>	Optional	<p>When an order is completed for this campaign, algorithms plugged into this spot are called to do additional work (e.g., create a customer contact). You need only specify this type of algorithm if you require additional work to be performed when an order is completed for this campaign.</p> <p>Click here to see the algorithm types available for this system event.</p>

Campaign - Package

A package controls the various types of service agreements that will be created if the customer selects the package. This page summarizes all packages that may be selected on orders associated with this campaign. Open this page using **Sales & Marketing, Campaign** and navigate to the **Package** tab.

Description of Page

Campaign is the user-defined code that uniquely identifies the campaign. The campaign's description appears adjacent.

The grid contains an entry for every package associated with this campaign. To view or change a package, press the adjacent go to button (which will transfer you to [Package - Main](#)). To add a new package, use the **Campaign** context menu to transfer to [Package - Main](#) in add mode.

Maintaining Packages

After a user enters basic information about a customer on an [order](#), they are shown the various packages that may be offered to the customer. A package controls the various types of service agreements that will be created if the customer selects the package.

If the customer elects to take a package, the system sets up / updates [the "V"](#) (along with all of the ancillary things that happen when service is initiated, e.g., field activities are created).

The topics in this section describe the package transaction.

Refer to [The Big Picture of Campaigns, Packages and Orders](#) for general information about packages. Refer to [The Big Picture Of Package Eligibility Rules](#) for a discussion of how to control which packages can be used for different types of customers. Refer to [Designing Campaigns and Packages](#) for guidelines describing how to design your packages.

An order can be completed without selecting a package. It is possible to use the order transaction to simply create / update persons and accounts. Refer to [Campaigns Without Packages](#) for more information.

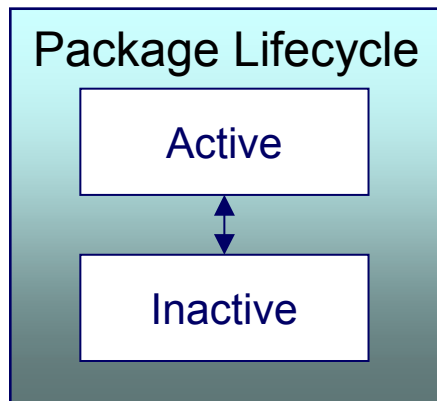
Contents

[Lifecycle of a Package](#)

[Package - Main](#)
[Package - SAs To Create](#)
[Package - Questions & Misc Fields](#)
[Package - Eligibility](#)
[Package - Instructions](#)

Lifecycle of a Package

The following diagram illustrates the lifecycle of a package.



Active

An **Active** package may be selected on an order.

Inactive

By **Inactivating** a package, you are preventing it from being selected on future orders.

Package - Main

This page is used to define basic information about a package.

Refer to [The Big Picture of Campaigns, Packages and Orders](#) and [Designing Campaigns and Packages](#) for more information.

Open this page using **Sales & Marketing, Package**.

Description of Page

Package contains a concatenation of basic information about the package. **Package ID** is the system-assigned unique identifier of the package. These values only appear after the order is added to the database.

Use **Campaign** to define the campaign to which this package belongs.

Use **Description** and **Long Description** to describe the package.

Status defines the state of the package. Refer to [Lifecycle of a Package](#) for information about the valid values and the state transition rules.

The tree at the bottom of the page summarizes the information that appears on the remaining tabs. You can click on a hyperlink to transfer you to the appropriate tab with the relevant information displayed.

Package - SAs To Create

This information on this page controls the type of service agreement(s) / service point(s) that are created if this package is selected on an order.

Refer to [The Big Picture of Campaigns, Packages and Orders](#) and [Designing Campaigns and Packages](#) for more information.

Open this page using **Sales & Marketing, Package** and navigate to the **SAs To Create** tab.

Description of Page

Package contains a concatenation of basic information about the package. **Package ID** is the system-assigned unique identifier of the package. These values only appear after the order is added to the database.

Warning! This following information is not intuitive; we strongly recommend that you follow the guidelines under [Designing Campaigns and Packages](#) before attempting to enter this information.

The **SAs To Create** scroll contains one row for every service agreement that will be created if this package is selected on an order. The following fields are used:

- **CIS Division** and **SA Type** define the [type of service agreement](#) to be created.
- **Start Option** controls how the system will populate the new service agreement with default values. Refer to [start options](#) for more information.
- **SP Type** defines the type of service point that will be created / reused. Refer to [Designing SP Types](#) for more information.
- **Start Method Algorithm** controls how the system creates the new service agreement and service point. Refer to [Determine The Method Used To Create Service Agreements](#) for more information.
- **Create Proposal SA** controls if the new service agreement is a real or a proposal SA. This switch only appears if the quotation **Contract Management** module is not [turned off](#). Refer to [Proposal Service Agreements Must Exist](#) for more information.

Package - Questions & Misc Fields

It is possible to design a campaign so that the user is prompted for additional information when a package is selected on an order. This page is used to define how the package overrides the questions and miscellaneous fields defined on the [campaign](#).

Refer to [The Big Picture of Campaigns, Packages and Orders](#) and [Designing Campaigns and Packages](#) for more information.

Open this page using **Sales & Marketing, Package** and navigate to the **Questions & Misc Fields** tab.

Description of Page

Package contains a concatenation of basic information about the package. **Package ID** is the system-assigned unique identifier of the package. These values only appear after the order is added to the database.

Campaign contains a concatenation of basic information about the campaign.

Warning! The following information is not intuitive; we strongly recommend that you follow the guidelines under [Designing Campaigns and Packages](#) before attempting to enter this information.

The grid contains one row for every [Questions & Misc Field](#) defined on the **Campaign**. You need only modify this information if this package behaves differently than the campaign (e.g., a field is required on the package, but optional on the campaign). To override a field on this package, turn on the **Override** switch and change the appropriate fields. There are several restrictions in respect of what can be overridden as described below:

- If a question / miscellaneous field is **Required** on the campaign, no fields may be overridden on the package.
- If a question / miscellaneous field is **Optional** on the campaign, you may override all field values. However, if the package has been used on a **Completed** order, you may not change its **Applicability** to be **Required**.
- If a question / miscellaneous field is designated as **only applicable on package** on the campaign, you should designate its applicability as **Required**, **Optional** or **Not Used On This Package**. If the field is either **Required** or **Optional** on this package, you may also override the other field values. Please note that if the package has been used on a **Completed** order, you may not change its **Applicability** to be **Required**.

Please refer to the [Description of Page](#) that appears under [Campaign – Questions & Misc Fields](#) for a description of the fields on the page.

Package - Eligibility

This page is used to define the conditions under which a package can be selected on an order.

Refer to [The Big Picture Of Package Eligibility](#) and [Determine Package Eligibility Rules](#) for more information.

Open this page using **Sales & Marketing, Package** and navigate to the **Eligibility** tab.

Description of Page

Package contains a concatenation of basic information about the package. **Package ID** is the system-assigned unique identifier of the package. These values only appear after the order is added to the database.

Warning! The following information is not intuitive; we strongly recommend that you follow the guidelines under [The Big Picture Of Package Eligibility](#) and [Determine Package Eligibility Rules](#) 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 package can be selected on an order (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 **Eligible** if this package can be selected on the order.
 - Choose **Ineligible** if this package cannot be selected on the order.
 - Choose **Check Next Group** if the next criteria group should be checked.
- 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 **Eligible** if this package can be selected on the order.
 - Choose **Ineligible** if this package cannot be selected on the order.
 - Choose **Check Next Group** if the next criteria group should be checked.

The grid that follows contains the package'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 package if their customer class is residential. 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 **Field** if you want to compare a response to a question / miscellaneous field to a given value. Push the adjacent search button to select the field.
 - Choose **Algorithm** if you want to compare anything other than a response to a question / miscellaneous field. 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 Type** using a logical operator. The following options are available:
 - Use **>, <, =, >=, <=, <>** (not equal) to compare the **Criteria Field Type** using standard logical operators. Enter the comparison value in the adjacent field.
 - Use **In** to compare the **Criteria Field Type** 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 Type** 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.
- 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 package will be judged **Ineligible** or **Eligible** 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 package will be judged **Ineligible** or **Eligible** 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 package will be judged **Ineligible** or **Eligible** based on the values defined above in **If Group is False** and **If Group is True**.

Package - Instructions

This page is used to define special instructions that should be presented to the user when they select this package. This information appears on the [Order - Package Confirmation](#) page.

Refer to [Determine If Additional Instructions Are Needed](#) for more information.

Open this page using **Sales & Marketing, Package** and navigate to the **Instructions** tab.

Description of Page

Package contains a concatenation of basic information about the package. **Package ID** is the system-assigned unique identifier of the package. These values only appear after the order is added to the database.

The grid contains a row for each instruction to be presented to the user when they select a package. The following fields are used:

- Use **Sort Sequence** to control the relative order of the instruction.
- Use **Display Icon** to control the icon that prefixes the instruction. Refer to [Display Icons](#) for more information.
- Use **Description** to define the text of the instruction.