IBM Cúram Social Program Management
Version 7.0.0

Cúram Financials Guide

IBM

Contents

Figures

Tables

Cúram Financials Guide

Cúram financials are used to process payments and liabilities that are received from persons and employers. A nominee is required for a financial to be processed. Payments can be captured, invalidated, canceled, or reissued. Liabilities can be reversed and written off.

Introduction

Purpose

The purpose of this guide is to provide a high-level overview of financial processing. After reading this guide, the reader should have a basic understanding of the financial building blocks and of how these building blocks are used to process payments, liabilities, and payments received from persons and employers.

In order to best understand these concepts, the guide should be read in full. The guide is not intended to be used as a training manual or user guide.

Audience

This guide is intended for readers with an interest in understanding the business concepts associated with financial processing within the application.

Prerequisites

As payments and bills are issued in relation to eligible cases, a basic understanding of case processing is required (see the Cúram Integrated Case Management Guide). The Cúram Nominees Guide and Cúram Deductions Guide are optional reading prerequisites.

Note: The ability to integrate the application Financials with an ERP financial system is provided. In an integrated environment, some of the processing described in this business guide occurs in the ERP financial system. For further information, please see the Cúram Financial Adapter Technical Overview Guide. For detailed information on configuring application's financials with the ERP system, please consult the Cúram Integrated ERP Operations Guide.

Chapters in this Guide

The following list describes the chapters within this guide:

Financial Building Blocks
> This chapter provides information on the building blocks of financial processing. These include financial components, financial instruction line items, financial instructions, and financial instruments.

Generating Payments and Liabilities
> This chapter provides information on generating payments and liabilities.

Processing Deductions and Adjustments
> This chapter provides information on processing deductions, tax adjustments, and surcharge adjustments.

Maintaining Payments and Liabilities
> This chapter provides information on maintaining payments and liabilities.

This includes information on canceling and reissuing payments, approving suspended payments, and reversing and writing-off liabilities.

Processing and Maintaining Payments Received
This chapter provides information on processing and maintaining payments received. Payments received can be recorded on the system and allocated toward outstanding liabilities.

Financial Instruction Types
This appendix provides an overview of each financial instruction type. It describes the potential instruction line item types that can get rolled-up into the financial instruction. It also describes any ILIs related to or that can be applied to the financial instruction as part of set financial processing (as described throughout this guide).

Financial Building Blocks

Financial components, instruction line items, instructions, and instruments are building blocks for Cúram financials. They are used to process outgoing payments and liabilities, to receive payments into the system, and to complete other account maintenance tasks.

Introduction

This chapter provides general definitions of the "building blocks" of financial processing. The building blocks include financial components, financial instruction line items, financial instructions, and financial instruments. They are used to process outgoing payments and liabilities, to receive payments into the system, and to perform other account maintenance tasks.

Payments, Liabilities, and Payments Received

Payments, liabilities, and payments received are the basic financial units. A payment is an issue of funds from the organization to a participant. Payments are issued to participants that are eligible for a benefit product. For example, a payment might be a $300 check issued to a person eligible for child allowance payments. Payments are also generated for participants when the organization has paid them too little or billed them too much. For example, if the person above should have been paid $325, an under payment is created in order to pay the person the additional $25 due.

A liability is a charge for funds from the organization to a participant. Liabilities are issued to participants that are eligible for a liability product. For example, a liability might be an invoice for $100 issued to an employer liable for service fees. Liabilities are also generated for participants when the organization has paid them too much or billed them too little. For example, if the employer above should have been billed $125, an under billing is created in order to bill the employer the additional $25 due.

A payment received is an amount of money received by the organization and recorded on the system. Payments received are usually sent to the organization in response to a bill. That is, they are used to pay off liabilities. Unlike payments and liabilities, payments received are not generated as part of case processing. Instead, they are recorded on the system either manually by a user or via a batch process that records payments received in bulk, for example, all payments received by EFT from a particular bank. Payments received are associated with a person's financials through the allocation of the payment received toward one or more outstanding liabilities.

Financial Components

Financial components are schedules to pay or bill in relation to a case. Case components are the benefits or liabilities that a primary client might be eligible for, e.g., personal benefit. A financial component is generated from a case component if the primary client is determined to be eligible for the case component. Each financial component contains the following information:

- Category - this setting determines whether the financial component is a payment for a benefit or a bill for a liability.
- Case nominee - the participant who will receive the payment or bill.
- Primary client - the client for whom entitlement (the amount to be paid/billed) was calculated.
- Amount to be paid or billed.
- Delivery pattern information including:
  - Delivery method and frequency, e.g., by cash on a weekly basis, by check on a monthly basis.
  - Cover period type and cover period offset, e.g., issue N number of days in advance where N is the cover period offset.
- Start date and end date for the payment or billing schedule.
- Scheduled due date for the payment/bill.

A sample financial component might indicate that John Smith should receive $25 per week in payments by check on a Friday from January 1st through April 1st. The payment schedule may have a cover period of issue in three days advance. Each weekly payment will have a due date on a Tuesday before the Friday, allowing for three days for the check to clear by Friday.

Financial components are also created as part of reassessment processing. That is, financial components are created for any over payments and under payments that have been issued on a case. For example, if John Smith is originally paid $25, but a change in evidence makes him eligible for $40, a financial component with an amount of $15 is created to rectify the under payment. (For more information, see "Over and Under Payment Processing" on page 25.)

Once-off and Recurring Financial Components

Financial components are either once-off or recurring.

Once-off financial components are processed once. An example of a once-off financial component is an under payment that is processed completely on a single date.

Recurring financial components are processed repeatedly according to their delivery frequency. Each time a recurring financial component is processed, the next due date is calculated by adding the nominee's delivery frequency to the current date. The next due date continues to be 'rolled forward' until the financial component reaches its end date.

Typically, recurring financial components are processed multiple times. Note, however, that recurring financial components will only be processed once if their lifespan is shorter than or equal to the length of their delivery frequency.

Secondary Financial Components

Secondary financial components are schedules to make deductions from payments. They are created when deductions are set up for benefit cases. Secondary financial

components are processed when case payments are issued. A total deductible amount is calculated for each nominee who is receiving a payment on the case. This is the total amount that can be deducted from a payment issued to the nominee. During deduction processing, each deduction is processed against the reducing balance of the total deductible amount calculated for the nominee.

Case Nominee and Preferred Delivery Pattern

Financial components are issued to the nominee assigned to a case component and delivered according to the nominee's preferred delivery pattern. The default case nominee for all case components is the primary client; however, a case nominee can be any individual or party designated to receive a case component on behalf of the primary client.

The nominee assigned to a case component can indicate a preferred delivery pattern. A delivery pattern defines the frequency and method by which payments or bills are issued. For example, a delivery pattern may be weekly by check on Mondays or daily by invoice. If a delivery method for a delivery pattern is check, the case nominee assigned to receive the check must have an address recorded in the system. Similarly, if the delivery method is EFT, he or she must have a bank account recorded in the system.

Note: Any currency with an active currency exchange rate can be assigned to a case nominee at the case level. The assigned currency will be used for all case payments/bills for case components to which the case nominee is assigned. A case nominee's currency can be changed over the life cycle of the case.

Cover Period Patterns and Offsets

A delivery pattern also defines the cover period pattern for payment or bill delivery. A cover period pattern specifies how payments or bills are issued, e.g., in advance, in arrears, once-off, etc. For example, the delivery pattern, 'weekly by check on Mondays', with a cover period pattern of 'in advance', indicates that each payment will be made on a Monday and will cover the week that starts on the Monday and continues to the next Sunday.

An offset typically defines the number of days in advance that a payment or bill needs to be processed in order to reach a case nominee on time. For example, check payments might be processed three days in advance so that the nominee receives the check on time. During financial component processing, the offset is used to calculate the due date of the financial component.

There are two types of offsets: the delivery method offset and the cover pattern offset. The delivery method offset allows for the time it takes to process and print certain delivery methods. For example, it may take the organization 2 days to process and print check payments. The cover period offset allows for the time it takes to deliver payments or bills by a specific delivery pattern. The cover pattern offset can be set for a delivery pattern when the selected cover pattern includes an offset. For example, Issue in advance - N days prior to issue; where N is the cover period offset that needs to be specified. For example, a two-day offset may be required for the weekly by check delivery pattern. This allows for the time it takes for a check to be sent through the mail.

If both offsets are used, they will be added together during financial component processing to calculate the due date. For example, if the delivery method, check, has an offset of 2 days and the delivery pattern, weekly by check, also has an offset of 2 days, then payments will be due 4 days in advance.

Note: Offsets are set at the product level as part of financial administration. They can also be modified as part of case creation.

Payment Exclusion Dates for Delivery Methods

Payment exclusion dates represent the days on which the organization will not be in a position to make payments using a particular delivery method. A prepayment requirement specifies that financial processing should occur on the nearest processing date prior to the exclusion date.

For example, if cash payments for benefit cases are normally made on a Monday and next Monday is a bank holiday then next Monday can be marked as an exclusion date on the cash payment financial calendar. Payment will then be made on the nearest valid processing date prior to the exclusion date.

A separate financial calendar is provided for each delivery method used by the organization. This is useful because different delivery methods may have different exclusion dates. For example, it may not be possible to issue checks on public holidays, but it may still be possible to process EFT transactions.

Prepayment works in addition to cover period offsets. For example, a Tuesday payment date is processed on the Friday in advance due to a two-day in advance cover pattern offset and payment exclusion date settings for the Saturday and Sunday.

Note: Exclusion dates are not applicable to online payments as these are front office payments that are a specific payment for a specific case on a specific date.

Financial Instruction Line Items and Financial Instructions

A financial instruction line item (ILI) is the most basic financial processing element within the application. At least one instruction line item is generated for every financial process that occurs. Instruction line items for a case are created when financial components are processed. Whereas a financial component is a schedule to pay or bill, an instruction line item for a case is an instance of that schedule, i.e. a payment or bill.

For example, a financial component schedules for John Smith to be paid $25 by check every week, starting on January 1st. On January 1st, the financial component is actually processed into a financial instruction line item (a financial instruction line item would thereafter be created every week until the end date of the financial component). Information carried over from the financial component includes the nominee, the delivery pattern, the amount, and the period of time the payment or bill covers.

A financial instruction is a rolled up view of one or more instruction line items. The rolling up of instruction line items into a single financial instruction consolidates a nominee's instruction line items of the same category into a single financial instruction. Rolling up may also be used to consolidate a number of payments or liabilities for a nominee.

For example, an instruction representing four separate payment instruction line items may be created and issued to a nominee as a single payment. Rolling up is useful because it simplifies the organization's financial interactions with participants. For example, a payment instruction might include the maximum personal instruction line item rolled up with any deduction instruction line items, e.g., third party deductions.

A payment group can also be created to allow the payment instruction line items from certain products/programs to be rolled up into one payment instruction. For example, if a nominee is paid $50, $75 and $100 from three different programs and no payment groups have been defined, all the payment instruction line items will be rolled up into a single payment instruction of $225. However, if an agency requires payments from certain programs to be rolled up separately, this can be achieved by creating payment groups. If a payment group is created for the programs/products of the first two payments ($50 and $75), then the system will generate two payment instructions - one for $125 and another for $100.

Financial instructions and related instruction line items can also be created outside of case processing. For example, they are created when a payment received is allocated toward a liability, when a manual payment is captured, and when a liability is written off. "Financial Instruction Types" on page 30 provides an overview of each financial instruction type and its related financial instruction line items.

Note: If your organization is using an integrated environment, instruction line items are the financial integration point between the two systems. In this environment, the remaining financial building blocks described in this chapter are not produced by the application, instead their equivalents are created and managed by the ERP financial system. For more information, please see the `Cúram Financial Adapter Technical Overview Guide`.

Information Used to Roll Up Financial Instruction Line Items

The following information is used to determine which financial instruction line items are to be rolled up into a financial instruction:

* Financial instruction line items of the same category can get rolled up into the same financial instruction. For example, when taxes are applied to a benefit, both the benefit instruction line item and the tax instruction line item will have the same category of benefit so that the related instruction line items can be rolled up and processed together.
* A financial instruction is issued to a single nominee; therefore, all instruction line items must share the same nominee.
* In order to be able to process a financial instruction, all instruction line items must have the same delivery method and currency.
* The financial instruction line item cover period is used to determine which financial instruction line items are ready to be processed at the same time and thus can be rolled up into one financial instruction.
* Each financial instruction line item has an amount which is either a credit or a debit and this amount is used to calculate the overall value of a financial instruction. For example, a payment instruction which includes a payment for $100 and a tax deduction of $10 would have a payment amount of $90.

Understanding the Difference between the Due Date and Processed Date

All financial instructions have a due date and a processed date. The due date is the date on which a payment or invoice is scheduled to be received by the nominee. For example, in the case of an electronic funds transfer, the due date is the date that the payment is scheduled to be deposited in the nominee's bank account and available to be withdrawn. As part of financial component processing, the due date is calculated using the nominee's delivery pattern. For example, if a client is paid every week on a Monday in advance, their due date would always be a Monday, that day being the day they are always due to be paid or invoiced.

The due date should not be confused with the actual date the nominee gets paid, which is known as the payment date. This is because in order to get payment to the nominee in time for the due date, any payments or invoices need to be processed before the financial processing date closes. For example, the financial processing date for the above delivery pattern closes on a Thursday (this is the offset period required to ensure the money can get to the nominee's bank account by the following Monday). On Wednesday, the client notifies a change in their circumstances which took effect from the Monday of this week. This change causes a reassessment which results in the client being underpaid. The underpayment covers the period from Monday to Friday of this week (because the client is paid weekly in advance). The amount that the client is underpaid is actually 'due' on the Monday at the start of this week. However, because this date has already past, the client can't actually receive this payment on this date.

However when changes occur that impact when payment can physically be made, it is important to note that the 'processed date' must also be considered when viewing financial transactions. The casework can use the processed date to reconcile that a due date may not have been met for the payment, where the processed date does not allow enough time for the due date to still be met. Taking the above example where the due date is Monday 13th November, but the actual processed date is Wednesday 15th November (which falls in time for the financial processing of the following Monday's payment), it reveals that payment date will be the Monday, 20th November.

Financial Instruments and Payslips

Financial instruments are the records of actual payments, liabilities, or payments received that have been issued or received by the organization. Financial instruments are recorded for financial instructions. For example, a payment instrument is recorded when a payment instruction is issued for a person.

A payslip is a physical record of the financial instruction line items that pertain to a particular payment or liability instruction used for record keeping purposes. They can be issued to a participant, case nominee, or third party. Each payslip contains a header with recipient details, payment or liability details, and a section containing instruction line item details for each instruction line item in the payment or liability.

Generating Payments and Liabilities

Payments and liabilities due for active cases are automatically generated when the financial batch suite is run. The financial batch suite is a group of four processes that run in sequence. Running these processes at regular intervals automates payment and liability generation and ensures that eligible participants are paid or billed according to their delivery frequency.

Overview

The diagram below provides a schematic overview of the processes within the Financial Batch Suite. The Determine Product Delivery Eligibility batch process runs first to generate financial components. The Generate Instruction Line Items batch process runs second to generate instruction line items. The Generate Instruments batch process runs third to generate financial instructions and financial instruments. The Generate Payslips process runs fourth to generate payslips for all financial instruments.
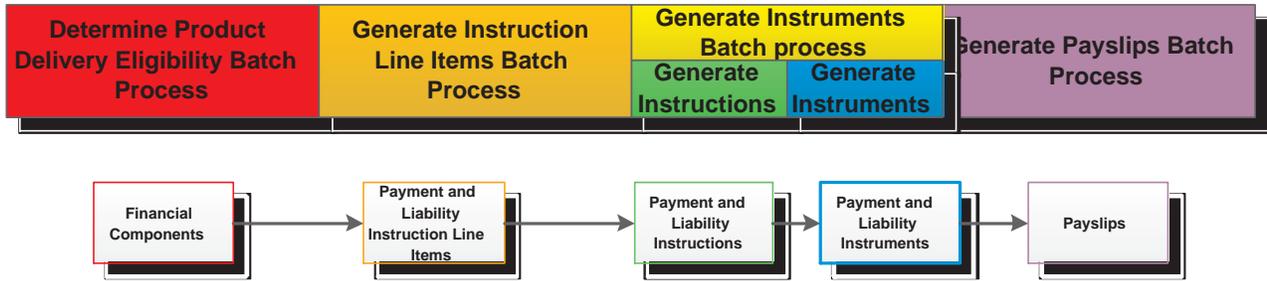
Figure 1. The Financial Batch Suite

> **Note:** If your organization is using an integrated environment, the steps described in the diagram above are not all applicable. Once the Generate Instruction Line Items batch process has been run, a new batch job, Transfer Instruction Line Items, is run to transfer the instruction line items to the ERP financial system for further processing. For more information, please see the `Cúram Financial Adapter Technical Overview Guide`.

Simulating Payments

Payments can be simulated online by a user. Simulating a payment allows a user to view all payments due to a participant for a specific date before the payment is generated and issued. This allows the user to see how much money a participant is due to receive over a given period. The system also displays any deductions due on payments. Simulating deductions on payments optimizes payment processing. If the user is not satisfied with the details of a deduction that is due to be processed, the deduction can easily be modified to achieve the desired result.

When a payment is simulated, the system retrieves the financial components created when the case was activated, or last reassessed, and rolls them forward until the date entered by the user is reached. It then generates a simulation of the payment for that week or for the delivery period of the case. The system also calculates the amount, cover period, and effective date of the payment. If deductions are set up on the case, these deductions are applied to the financial components and the reduced payment amount is also displayed. If no deductions exist on the case, the system only displays the payments due on that particular date.

A history of simulated payment records is maintained over time, allowing agency workers to provide a holistic view of potential payments with deductions to clients.

Creating Financial Components

When the Determine Product Delivery Eligibility process is called for a case (either through batch processing or online), eligibility is determined for each case component, e.g., personal benefit, child benefit. Financial components are created from eligible case components using the financial component's nominee delivery pattern. Secondary financial components are also created for every deduction that has been set up on the case.

The creation of financial components is handled by the Cúram Eligibility and Entitlement Engine (see the `Inside Cúram Eligibility and Entitlement Using Cúram Express Rules` guide).

Expiring Financial Components

A financial component is expired when it reaches its end date. For once-off financial components, this means that expiration occurs immediately after the financial component is processed. For recurring financial components, the financial component remains live, and continues to be processed, until the end date is reached. It is then expired.

A financial component will also be expired if reassessment finds the primary client to be ineligible for a case component linked to the financial component. This may occur any time the financial component is reassessed and ensures that a financial component will cease to be processed if a change in circumstances affects a primary client's eligibility.

Generating Financial Instruction Line Items

Financial components are processed into financial instruction line items when the Generate Instruction Line Item batch process is run. The financial components for a single benefit case can also be processed into instruction line items as part of generating payments online. Reassessment is always performed before a financial component is processed. This ensures that the primary client is still eligible and that the financial component is still valid. A single financial instruction line item is generated when a once-off financial component is processed. In contrast, a financial instruction line item is generated every time a recurring financial component is processed.

Before processing a financial component into an instruction line item, the system checks for any payment exclusion dates set up for the delivery method. If found, the processed date for the instruction line item will be adjusted accordingly. For example, if the delivery method is cash and the financial component due date is on a payment exclusion date for the cash delivery method, the system will adjust financial instruction line item processed date to a valid processing date.

Note: If your organization is using an integrated environment, after the instruction line items have been generated and transferred, the remaining processing is carried out by the ERP financial system. For more information, please see the Cúram Financial Adapter Technical Overview Guide.

Generating Financial Instructions

Financial instructions for benefit and liabilities are created when the Generate Instructions batch process runs as part of the Generate Instruments batch process. The financial instructions for a single benefit case can also be created as part of generating payments online.

Payment and liability financial instructions are created when one or more instruction line items are rolled up into a single financial instruction. Note that only instruction line items with the same category, nominee, delivery pattern, and currency can be rolled up together. For example, two payment instruction line items generated from the same financial component over two processing dates may be rolled up into a payment instruction. Or, a payment instruction line item may be rolled up with a deduction or tax adjustment line item (see "Processing Deductions and Adjustments" on page 13). However, if the Agency does not wish to roll up all the payment instruction line items from different programs into a single instruction, they can do so by adding programs/products to a payment group. This will ensure that only payment instruction line items belonging to a payment group will be rolled up into a single instruction.

For example, a nominee is paid $50, $75 and $100 from three different programs. If a payment group has not been defined, all the payment instruction line items will be rolled up into a single payment instruction of $225. However if it is required that the payment of $50 from a particular program should not be rolled up with payments from the other programs, this can be achieved by creating a payment group and adding the program/product whose payment should not be rolled up with payments from other programs. By doing so, the system will generate two payment instructions - one for $50 and another for $175.

A payment or liability instruction is created when the first instruction line item to be rolled up into the instruction is processed. Depending on the type of the remaining instruction line items, the amount of the instruction is then added to or subtracted from. For example, if a payment instruction line item is rolled up into a payment instruction, the amount of the payment instruction would be lessened when a related deduction instruction line item is rolled up into the same payment instruction.

When a payment instruction line item is rolled up, its outstanding amount is updated to zero and its status is changed to processed. These changes represent the fact that the instruction line item has been processed into a payment instruction and that a payment has been issued to the case nominee.

When a liability instruction line item is rolled up, its status is also changed to processed. Note, however, that its outstanding amount remains equal to the full amount of the liability. This indicates that the liability is still outstanding and that the amount is still owed to the organization. The outstanding amount will only be reduced or cleared when the liability is written off, reversed, or allocated against (see "Maintaining Payments and Liabilities" on page 21).

Generating Financial Instruments

Financial instruments are generated when the Generate Instruments batch process is run. This process is run directly after the Generate Financial Instruction batch process.

To allow for integration with external financial systems, payment/billing information from financial instrument records is extracted to an external file. For example, payment information for instruments with a delivery method of "check" would be extracted to a system used to draft checks.

Generating Payslips

If payslips are required, the Generate Payslips batch process can be run following the generation of instruments. The type of payslip generated depends on the type of recipient, e.g., the primary client, a utility.

Issuing Payments Online

Payments can be issued online for a single benefit case without the Financial Batch Suite having to be called. This may be necessary if the primary client is in need of emergency assistance. Payments can only be issued online for active cases.

Note: If your organization is using an integrated environment, issuing payments online is coordinated across both systems. For more information, please see the `Cúram Financial Adapter Technical Overview Guide`.

Issuing Payments to Third Parties

Payments can be issued to third parties, e.g., utility payments can be issued to utility participants. Third-party payments are processed according to the third-party participant's preferred payment method, payment frequency, and next payment date. The payment method defines the method of payment, e.g., cash, check, etc. The payment frequency defines the frequency with which payments will be issued. The next payment date is calculated by adding the payment frequency to the previous payment date. A preferred currency can also be set for each participant. This setting defines the currency in which third-party payments will be issued to the participant.

Third-party payments are issued when the Issue Concern Payments batch process is run. This batch process searches for all participants due for payment. It processes payments for each of these participants and issues payments according to the payment method defined for each. As part of issuing concern payments, the system checks for any payment exclusion dates set up for the third party participant's delivery method and adjusts the payment date according. (For more information on payment exclusion dates, see "Payment Exclusion Dates for Delivery Methods" on page 5.)

The next payment date is then rolled forward to the next valid payment date. For example, a utility participant may specify that it wishes to receive payments from the organization every quarter by electronic fund transfer (EFT). This information would be stored on the utility participant record. Every quarter, a payment would be issued by EFT. The next payment date would then be rolled forward to the appropriate date in the next quarter.

Sample Payment Generation

This section provides an example of payment generation. In this example, the primary client (and case nominee), John Smith, is eligible to receive a $35 payment once a week for one month. The payments are issued for the case component, Maximum Personal Benefit. The following sections illustrate the processing that would occur at each stage of financial processing.

Creating Financial Components

When the Determine Product Delivery Eligibility batch process is run, John Smith is found eligible for the Max Personal Benefit from February 1st, 2005 to February 28th, 2005. A recurring financial component is created with the following details:

- Case nominee = John Smith
- Amount = $35
- Delivery method = Check
- Delivery frequency = Weekly in advance
- Category = Payment
- Type = Max Personal Benefit
- Start Date = February 1
- End Date = February 28
- Expiry Date = February 21
- First Due Date: February 1

Generating Financial Instruction Line Items

On February 1st, the Generate Instruction Line Items batch process runs and the system searches for any financial components with a processing date on or before

February 1st. John Smith's financial component has reached its first due date, so an instruction line item is created. The information stored on the instruction line item includes the following:

- Case nominee = John Smith
- Delivery method = Check
- Category = Payment
- Type = Max Personal Benefit
- Cover period = February 1 to February 7
- Amount = $35
- Unprocessed amount = $35
- Status = Unprocessed
- Processed Date: February 1

After the instruction line item is created, the due date on the financial component is rolled forward to February 8th.

Generating a Financial Instruction

When the Generate Instruments batch process is run, the instruction line item for February 1st to February 7th is rolled up into one payment financial instruction.

The following describes the payment financial instruction:

- Case nominee = John Smith
- Delivery method = Check
- Category = Payment Instruction
- Total Amount = $35
- Cover Period = February 1 to February 7

At this point, the status of the payment instruction line item is updated to "processed" and the outstanding amount is updated to zero.

Generating a Payment Instrument

When the Generate Instruments batch process is run, a payment instrument is created from the payment financial instruction. This instrument includes all of the information listed above for the payment financial instruction.

Generating a Payslip

When the Generate Payslips batch process is run, a payslip can be generated for the payment financial instruction. The payslip includes the following information:

- Nominee Name = John Smith
- Component Type = Max Personal Benefit
- Cover Period From: February 1
- Cover Period To: February 7
- Amount = $35

The Next Processing Dates

On February 8th, another instruction line item is created with a cover period of February 8th to February 14th. The next processing date is then rolled forward to February 15th.

A financial instruction, instrument, and payslip is created for the new instruction line item.

This processing will repeat on every processing date until the recurring financial component is expired.

Processing Deductions and Adjustments

Deductions, tax, and surcharge adjustments are processed when payments and liabilities are generated. Deductions and tax adjustments are applied to benefits while surcharge adjustments are applied to liabilities. Financial account adjustments can be made to a participant's financials account.

Introduction

This chapter provides information on processing deductions and adjustments. The last two sections in this chapter provide samples of a payment generation with deductions and tax adjustments and of a liability with a surcharge adjustment.

Deductions

Deductions allow the organization to allot part of a person's benefit payments to a specific purpose. This purpose is dependent on the category of deduction. There are three deduction categories within the application: applied deductions, un-applied deductions, and third party deductions.

For applied deductions, the amount is deducted from the benefit and applied toward any participant's outstanding liability. For example, $10 of a person's weekly benefits can be applied toward paying off an overpayment that was previously issued to him or her. A payment can be reissued with an applied deduction if it is cancelled.

For un-applied deductions, the amount is also deducted from the benefit. It is then saved on the system as an unprocessed instruction line item payable to the organization. Un-applied deductions are used to recoup monies previously paid out by the organization. For example, if a person was previously issued money out of an emergency fund, deductions could be made from the person's benefit payments as a means of recouping the money for the organization. A payment can be reissued with an un-applied deduction if it is cancelled.

Third party deductions allow a portion of a person's benefit to be directed toward bills, charges, or debts owed to a third party. Third party payments are made to any other participants registered on the system. For example, a person can use a portion of a benefit to pay his or her electricity bills. The electricity provider is a registered utility on the system.

Deduction are set up as part of case administration. A complete history of deductions, active and inactive, is maintained at the case level. For detailed information on setting up and maintaining deductions over time, see the Cúram Deductions Guide.

Creating Secondary Financial Components for Deductions

Once the primary financial components have been created, secondary financial components are created for each deduction existing on the case. If a deduction has been created against a particular benefit component, the deduction financial component is associated with the benefit financial component.

Generating Instruction Line Items for Deductions

Financial instruction line items are generated for all three deduction categories (applied, un-applied, third party) by the processing of secondary financial

components. Secondary financial components are processed as part of the Generate Financial Line Items batch process. This occurs after all payment instruction line items have been generated, i.e., after the primary financial component has been processed. The amount of the deduction is checked against settings that are configured as part of system administration. Based on this comparison, the system determines whether or not the secondary financial components are processed.

For example, before processing the secondary financial components, the system determines whether or not there are sufficient funds in the payment amount to cover the deduction amount. All deductions have a setting that determines the action to take if there are insufficient funds available to process the full deduction amount. Depending on this setting, part of a deduction may be processed against the amount that is available or the deduction may not be processed at all.

Deductions can be prioritized by the organization in order of importance. For example, it may be more important to process a deduction that is used to pay housing costs than a deduction that is used to repay a liability owed to the organization. Based on its priority, each deduction is applied to the total deductible amount calculated for the nominee who receives the benefit payment. For each applied, stand alone, or utility payment deduction that is processed, two line item records are created. The first of these is a debit against the benefit. As described in the following section, the second is applied toward a liability, stored on the system, or paid toward a utility, depending on its type.

Configuration is also available which provides the ability for the agency to define whether or not overlapping deductions are allowed. If a deduction is configured to prevent overlapping deductions, a validation will be displayed if a user tries to activate a deduction which already exists on the case for an overlapping time period. This can be configured for all categories of deductions (applied, un-applied and third party).

The two instruction line items created are related to each other by a relationship record that is automatically created by the system. This allows for traceability back to the case nominee whose payment the deduction was made from. For example, when a bulk payment is issued to a utility (see below), each line item in that payment will be traceable back to a case nominee.

Processing Deductions

The first instruction line item created for all deduction categories is rolled up into the payment instruction for the benefit. This means that the person in receipt of benefits will receive a payment less the deduction amount. For example, a utility deduction amount of $5 might be subtracted from a payment of $35 so that the payment instruction for the benefit would be for the total amount of $30.

For applied deductions, the second instruction line item will be rolled up into a payment received instruction that is allocated toward the outstanding liability. Note that the allocated payment received instruction is system-generated, but that it functions identically to an allocated payment received from outside the system (for more information on allocating payments received, see "Allocating Payments Received" on page 27).

For un-applied deductions, the second instruction line item is saved on the system as a line item payable to the organization.

For third party deductions, the third party payment instruction line item is issued to the relevant participant company by calling the Issue Concern Payments batch

process. This batch process is used to issue payments to participants. When run for a participant, it searches for all unprocessed third party payment instruction line items for the participant and rolls them up into one payment instruction. This allows the organization to issue a single payment to a participant in respect to deductions made from multiple cases over a period of time. For example, the organization may issue a utility a payment once per quarter.

For more information on deduction processing, see the Cúram Deductions Guide.

Tax Adjustments

Benefits can be adjusted to take relevant taxes into account. For example, a 10% tax may be deducted from every payment. A setting at the product level indicates whether adjustments are required for a product. If the setting is turned on for a benefit product, taxes will be applied to all payments issued in respect of the product.

The adjustment rate for taxes can be maintained as part of rate table administration. The same rate will be applied to all payments. For example, a tax of 5 percent may be applied to all payments for a benefit product. When financial components are created for a benefit, information regarding whether or not tax adjustment processing is required is saved on each of the financial components.

Generating Instruction Line Items for a Tax Adjustment

When the Generate Instruction Line Items batch process is run, or when payments are generated online, the system determines whether or not taxes should be applied to the benefit. If taxes should be applied, the system creates two instruction line items. The first of these is a debit against the benefit. As described in the following section, the second is used to pay the tax authority.

The two instruction line items created for a tax adjustment are related to each other by a relationship record. This allows for traceability back to the case nominee from whose payment the tax adjustment was made. For example, when a bulk payment is issued to a tax authority (see below), each line item in that payment will be traceable back to a case nominee.

Applying Taxes

The first instruction line item created for the tax adjustment is rolled up into the payment instruction for the benefit. This means that the person in receipt of benefits will receive benefits less the amount of the tax adjustment. For example, a benefit with a payment amount of $50 might be adjusted by a 10% tax rate so that the payment instruction for the benefit would be for a total of $45.

The second line item for the tax adjustment is issued to the tax authority by calling the Issue Concern Payments batch process (note that tax authorities are registered as service supplier participants). The system searches for all unprocessed tax payment instruction line items and rolls them up into one payment instruction for the tax authority. This allows the organization to issue a single payment to a tax authority for deductions made from multiple cases over a period of time. For example, the organization may issue a tax authority a payment once per tax year.

Surcharge Adjustments

Surcharges are additional charges applied to liabilities that have not been processed within an appropriate time period. For example, if an employer is issued

a bill and does not pay all or a portion of this bill within the appropriate time period, a surcharge is applied to the outstanding amount. The employer is issued a new bill for the surcharge.

Generating Instructions for Surcharge Adjustment

Every time the Generate Instruments batch process is run, the system searches for all existing instruction line items with outstanding amounts greater than zero. It then determines whether or not surcharge adjustment processing is required for each instruction line item and whether the instruction line item is due for surcharge processing. If these three factors are met, the system applies the surcharge rate to the unprocessed liability amount in order to determine the surcharge amount. An instruction line item is then created for the surcharge adjustment.

When the Generate Instructions batch process is run, the system searches for all surcharge instruction line items for the same client and rolls these instruction line items up with any additional liability instruction line items that the client holds. For example, if a surcharge instruction line item is created for an employer, this will be rolled up with any other liability line items existing for the employer when the Generate Instructions batch process is run.

A setting at the product level indicates whether adjustments are required for a product. If the setting is turned on for a liability product, surcharges will be applied to bills if they remain outstanding for one month. If a bill is not cleared within one month, it is surcharged at the adjustment frequency configured at the product level. The adjustment rate for surcharges is set at a fixed rate. For example, the organization may specify that liabilities left unpaid for one month will be surcharged at five percent.

Financial Account Adjustments

Adjustments can be used to correct a client's financial account so that the system does not attempt to generate a compensating over or under payment. For example, the organization might need to write a check manually to the client and can balance the client's account through an adjustment. A client's financial account can be adjusted by applying a debit or credit amount to the account. Adjustments do not result in any payment or liability that is issued to the client and are purely to balance an account.

Sample Payment Generation with Deductions and Tax Adjustments

The primary client (and case nominee), John Smith, is eligible to receive a $35 check once a week. This is for the case component, Max Personal Benefit.

John Smith is eligible for this payment from February 1st to February 7th (one week).

A third party deduction is also set up on the Max Personal Benefit case component with an amount of $5. This deduction is to be paid toward John Smith's electricity bills.

Additionally, tax adjustment processing is set up on the benefit.

Creating Financial Components

John Smith is found eligible for the Max Personal Benefit from February 1st, 2005 to February 7th, 2005. A financial component is created with the following details:

- Nominee = John Smith
- Amount = $35
- Delivery method = Check
- Delivery frequency = Weekly in advance
- Category = Payment
- Type = Max Personal Benefit
- Start Date = February 1
- End Date = February 7
- Expiry Date = February 1
- Processing Date: February 1
- Tax Adjustment = True

A secondary financial component for the third party deduction is created with the following details:
- Nominee = John Smith
- Amount = $5
- Delivery method = Check
- Delivery frequency = Weekly in advance
- Type = Deduction Payment
- Start Date = February 1
- End Date = February 7
- Expiry Date = February 1

Generating Payment Financial Instruction Line Item

On February 1st, the primary financial component reaches its processing date, and an instruction line item is created. The information stored on the instruction line item includes the following:
- Nominee = John Smith
- Amount = $35
- Delivery method = Check
- Category = Payment
- Type = Max Personal Benefit
- Cover period = February 1 to February 7
- Unprocessed amount = $35
- Status = Unprocessed
- Tax Adjustment = True

After the instruction line item is created, the financial component is expired, as there is no next processing date.

Generating Third Party Financial Instruction Line Items

When the primary financial component is processed into the payment instruction line item, the secondary financial component is also processed and a third party payment and a third party deduction instruction line item are created. The third party deduction instruction line item indicates the following:
- Nominee = John Smith
- Amount = $5
- Delivery method = Check

- Category = Payment
- Type = Deduction Item
- Cover period = February 1 to February 7
- Unprocessed amount = $5
- Status = Unprocessed

The third party payment instruction line item indicates the following:
- Nominee = The Electric Company
- Amount = $5
- Category = Third Party Payment
- Type = Deduction Payment
- Cover period = February 1 to February 7
- Unprocessed amount = $5
- Status = Unprocessed

Generating Tax Financial Instruction Line Items

The tax rate is set to 10%. A tax deduction and a tax payment instruction line item are created. The tax deduction instruction line item indicates the following:
- Nominee = John Smith
- Amount = $3.50
- Delivery method = Check
- Category = Payment
- Type = Tax Deduction
- Cover period = February 1 to February 7
- Unprocessed amount = $3.50
- Status = Unprocessed

The tax payment instruction line item indicates the following:
- Nominee = Tax Authority
- Amount = $3.50
- Delivery method = Check
- Category = Tax Payment
- Type = Tax Payment
- Cover period = February 1 to February 7
- Unprocessed amount = $3.50
- Status = Unprocessed

Generating a Payment Instruction

The payment instruction line item, third party deduction instruction line item, and tax deduction instruction line item for February 1st to February 7th are rolled up into one payment instruction.

The following describes the payment financial instruction:
- Nominee = John Smith
- Delivery method = Check
- Category = Payment
- Total Amount = $35 - $5 - $3.50 = $26.50
- Cover Period = February 1 to February 7

At this point, the status of the three instruction line items are updated to processed and their outstanding amounts are updated to zero.

Note: If your organization is using an integrated environment, payment instructions are not generated as described above. The equivalent processing is carried out by the ERP financial system. For more information, please refer to the `Cúram Financial Adapter Technical Overview Guide`.

Generating Payment Instructions for the Third Party and Tax Authority

The tax payment instruction line item and the third party payment instruction line item are rolled up into payment instructions for the tax authority and the electricity company, respectively. This is done when the Issue Concern Payments batch process is run. (Note that this batch process is usually run across many cases so that the third party and tax authority would receive single payments for all deductions made over a period of time). Note that the Issue Concern Payment batch process also generates payment instruments and payslips for the third party and tax authority, respectively.

Note: If your organization is using an integrated environment, payment instructions for a third party and the tax authority are not generated as described above. The equivalent processing is carried out by the ERP financial system. For more information, please refer to the `Cúram Financial Adapter Technical Overview Guide`.

Generating a Payment Instrument

A payment instrument is created from the payment instruction for the benefit when the Generate Instruments batch process is run. This instrument includes the information described above for the payment financial instruction.

Note: If your organization is using an integrated environment, payment instruments are not generated as described above. The equivalent processing is carried out by the ERP financial system. For more information, please refer to the `Cúram Financial Adapter Technical Overview Guide`.

Generating Payslips

A payslip can be generated for the payment instruction for the benefit. The payslip includes the following information:

*   Nominee Name = John Smith
*   Component Type = Max Personal Benefit
*   Cover Period From: February 1
*   Cover Period To: February 7
*   Amount = $26.50

Note: If your organization is using an integrated environment, payslips are not generated as described above. The equivalent processing is carried out by the ERP financial system. For more information, please refer to the `Cúram Financial Adapter Technical Overview Guide`.

Sample Liability with Surcharge Adjustments

This example provides a sample liability with surcharge adjustments. The primary client (and case nominee) is the employer, Midway Emporium. Midway Emporium is liable for service fees of the amount of $100 per month from January 2005 through February 2005.

Additionally, surcharge adjustment processing is set up on the liability. The surcharge rate is 10%. The time limit for paying the liability is 1 month; surcharges will be applied to liability instruction line item that go unprocessed for over 1 month.

Creating a Financial Component

When the Determine Product Delivery Eligibility batch process is run, Midway Emporium is determined to be liable for service fees in the amount of $100 per month from January 2005 through February 2005. A recurring financial component is created with the following details:

- Nominee = Midway Emporium
- Amount = $100
- Delivery method = Invoice
- Delivery frequency = Monthly in advance
- Category = Liability
- Type = Service Fee
- Start Date = January 1
- End Date = February 28
- Expiry Date = Feb 1
- Processing Date: January 1
- Surcharge Adjustment = True

Generating a Liability Financial Instruction Line Item

When the Generate Instruction Line Items batch process is run, the system searches for any financial components with a processing date on or before the current date. On January 1st, the financial component reaches its first processing date, and an instruction line item is created. The information stored on the instruction line item includes the following:

- Nominee = Midway Emporium
- Amount = $100
- Delivery method = Invoice
- Category = Liability
- Type = Service Fees
- Cover period = January 1 to January 31
- Unprocessed amount = $100
- Status = Unprocessed
- Surcharge Adjustment = True

After the instruction line item is created, the next processing date on the financial component is rolled forward to February 1st.

Generating a Liability Instruction

When the Generate Instruments batch process is run, the instruction line item for January 1st to January 31st is rolled up into one liability financial instruction.

The following describes the liability financial instruction:

- Nominee = Midway Emporium
- Delivery method = Invoice
- Category = Liability
- Total Amount = $100

- Cover Period = January 1 to January 31

At this point, the status of the liability instruction line item is updated to processed. The outstanding amount remains at $100 until allocations are made toward it.

Note: If your organization is using an integrated environment, liability instructions are not generated as described above. The equivalent processing is carried out by the ERP financial system. For more information, please refer to the *Cúram Financial Adapter Technical Overview Guide*.

Generating a Liability Instrument

When the Generate Instruments batch process is run, a liability instrument is created from the liability instruction. This instrument includes all of the above information for the liability instruction.

Note: If your organization is using an integrated environment, liability instruments are not generated as described above. The equivalent processing is carried out by the ERP financial system. For more information, please refer to the *Cúram Financial Adapter Technical Overview Guide*.

The Next Processing Date

On February 1st, another liability instruction line item is created with a cover period of February 1st to February 28th. The financial component is expired as this is the last processing date.

The system reads that no payments have been made toward the liability instruction line item for the month of January. Since it has been one month since the instruction line item has been issued, surcharges must be applied to the unprocessed liability instruction line item. As stated previously, the surcharge rate is set to 10%. A surcharge instruction line item is therefore created with the following details:

- Nominee = Midway Emporium
- Amount = $10
- Delivery method = Invoice
- Category = Liability
- Type = Surcharge
- Cover period = January 1 - January 31
- Unprocessed amount = $10
- Status = Unprocessed

The surcharge instruction line item and the liability instruction line item for February will be rolled up into a single liability instruction. A liability instrument will then be created.

Maintaining Payments and Liabilities

Payment and liability processing requires continuous maintenance to capture how client circumstances change over time. Maintenance functions for payments include the ability to capture manual payments, invalidate payments, cancel and reissue payments, and approve suspended payments. Maintenance functions for liabilities include the ability to reverse and to write off liability instructions.

Introduction

This chapter provides information on maintaining payments and liabilities. Maintenance functionality for payments includes the ability to capture manual payments, invalidate payments, cancel and reissue payments, and approve suspended payments. Maintenance functionality for liabilities includes the ability to reverse and write-off liability instructions.

Capturing Manual Payments

A manual payment is a payment to a case nominee that is issued outside of the system, such as when the system is off-line or when a manual payment is issued by an agency not normally connected to the system. If for any reason the system is unavailable, and a payment is issued to a person, then information regarding that manual payment can be captured on the system.

Capturing the manual payment calls processes that recreate all the elements of a payment including the payment instrument, the payment instruction, the rolled up instruction line items, and any relevant deduction/tax instruction line items.

The cover period of a manual payment can be recorded to facilitate the recording of a manual payment for a specific past or future cover period. If the cover period is specified the system will use that cover period in calculating the entitlement amount for a client.

After creating the financial elements of the manual payment, the system then processes the payment due so that the payment is not paid out a second time. The system then compares the amount of the manual payment to the amount of the processed payment and creates an over or under payment to rectify the difference. A notification task is also sent to the case owner notifying him or her that an over or under payment has occurred.

Note: If your organization is using an integrated environment, the capture manual payment process differs from the one described here. For more information, please refer to the `Cúram Financial Adapter Technical Overview Guide`.

Canceling and Reissuing Payments

A payment issued in error can be cancelled. Canceling a payment indicates that a payment has not been received. For example, a payment may be cancelled if a participant reports that a check has been lost in the mail. A payment is cancelled at the payment instruction level. When a payment instruction is cancelled, the status of the payment instruction changes from "issued" to "cancelled". All of the instruction line items within the payment instruction are also cancelled and their status changes from "processed" to "cancelled".

A payment may also be cancelled because the bank account has been closed or an error in instrument details may require the cancellation of the payment. When a payment is cancelled, the reason for canceling the payment is recorded. On cancellation, the payment instruction is negated rather than deleted from the system. This is for accountability and traceability purposes. A new reversal instruction is created to cancel out the amount of the payment instruction. A reversal instruction line item is also created for every instruction line item that was rolled up into the payment instruction.

Any payment that has been cancelled can be reissued to the original nominee or to an alternative nominee. For example, a check payment that has been lost can be

reissued to the original nominee. Payments can be reissued using any of the nominee's active delivery patterns. For example, an original payment issued monthly by check can be reissued daily by cash.

A payment that has been cancelled can be reissued and have an applied, or un-applied, deduction created against it. This creates a deduction, which is visible to a caseworker as per any other deduction on the system, including a full deduction history. A deduction created when reissuing a payment is processed into the reissued payment, and in the case of an applied deduction, applied against the overpayment. This achieves a reduced payment issued to the client for the cover period in question. Invalidating payment with a replacement payment, as described below, is an alternative mechanism to achieve a reduced payment to a client. Whether to use reissue payment with deduction applied or invalidate payment with replacement payment, depends on whether the change in circumstance is recorded before the payment is canceled. For example if a payment is issued to a client and the client dies during this period. If the agency is first notified of the death prior to payment being returned (or perhaps payment issuance later fails if, for example, bank account were closed), the date of death is recorded which generates a change in circumstance and hence an overpayment is created. At this point, to get a new payment issued for the period up to the date of death, Invalidate Payment cannot be used as the period has already been reassessed and therefore there is no longer a payment only for the cover period in question; the payment and the reassessment result (payment correction or over/under payment) need to be considered together for this period. In this scenario a reduced payment can be issued for the cover period by canceling the payment and reissuing with a deduction applied to the overpayment case.

When a payment is reissued, new payment instruction and payment line items are created. The cancelled payment instruction and its instruction line items have a status of "cancelled". The new payment instruction has a status of "issued", and the new payment instruction line items have a status of "processed". A payment can be reissued during the cancellation process or after the payment instruction has been cancelled.

Note: If your organization is using an integrated environment, the payment cancellation and reissuing process differs from the one described here. For more information, please refer to the `Cúram Financial Adapter Technical Overview Guide`.

Invalidating Payments

Payments issued to clients can be invalidated. The difference between invalidating and canceling a payment is that a canceled payment can be reissued. When a payment is invalidated, it will no longer be considered by system processing. This is to allow for the reissuing of a replacement payment if, for example, the original payment was issued for the incorrect amount. The payment can be both cancelled and invalidated at the same time. The system allows for this to happen at time of cancellation, or in two stages: cancel now and invalidate later.

For example, John Smith is receiving benefit payments of $70 per week, paid in advance on a Monday. He goes back to work on a Tuesday, which means he is only entitled to benefit on the Monday, i.e., he is only due $10 that week. He receives a check for $70 but phones the organization and informs it of his change of circumstance. The payment is cancelled, as he will not be cashing his check of $70, and invalidated, as the organization wants to issue a replacement payment.

The certification on his benefit case is modified to end John Smith's payments on the Monday. This results in a replacement payment for $10 being issued to him.

Furthermore, if there is a question mark over which day John returned to work when he informs the organization of his change of circumstance, the organization may require documentation detailing the date he started in his new employment. In this instance, the organization can cancel John's payment but not invalidate it until it receives the documentation from his new employer.

When a payment is invalidated, all of its associated components are invalidated with it.

Approving Suspended Payments

As part of application administration, a maximum amount can be set up for a delivery method, so that no payments over this amount can be made by the delivery method. Any payments that exceed this maximum amount will be suspended. For example, the organization may have a policy that no payments over $100 can be made by cash. If John Smith was scheduled to receive a payment for $120 in cash, the payment would be automatically suspended until the organization had an opportunity to investigate the reason that the limit was exceeded.

The case nominee will not receive a suspended payment unless the payment is approved by the organization. Approving a suspended payment overrides the maximum amount limitation and allows the payment to be issued. Over and underpayments are still created on cases with suspended payments regardless of whether the suspended payment has been approved or not. This is due to the fact that the over or underpayment may be related to several payments that were made over a period of time, of which some may not be suspended and are therefore valid. Also, the expectation is that agencies will deal with the situation in which a payment has been suspended in a timely manner; otherwise the client will not get paid. If the agency is not in a position to act on a suspended payment quickly or they do not want to create over or under payments on cases with suspended payments, they could choose to suspend the case itself.

Note: If your organization is using an integrated environment, the process for approving suspended payments differs from the one described here. For more information, please refer to the `Cúram Financial Adapter Technical Overview Guide`.

Reversing Liabilities

All or portions of liabilities can be reversed. For example, a liability instruction issued to a nominee in error can be reversed such that the nominee is no longer be responsible for the full amount owed on the liability. Alternatively a portion of a liability can be reversed, e.g., one liability instruction line item as part of a liability instruction can be reversed, such that the nominee is no longer responsible for the reversed line item but is still responsible for the outstanding amount on the liability instruction.

When a liability instruction line item is reversed, the liability amount is negated rather than deleted from the system. This is for accountability and traceability purposes. A new reversal instruction line item is created to cancel the amount of the liability instruction line item. The reversal instruction line item is automatically allocated toward the liability instruction line item. This updates the outstanding

amount of the liability instruction line item to zero. The reversal instruction line item is rolled up into a reversal instruction.

If the liability instruction line item was allocated toward before it was reversed, these allocations are once again made available to be applied toward other outstanding liabilities. In order to make this possible, the system automatically creates a reversal instruction line item for each allocation and rolls up the instruction line item or items into a reversal instruction. These reversal instruction line items can be allocated toward any of the nominee's remaining liabilities.

Note: If your organization is using an integrated environment, the liability reversal process differs from the one described here. For more information, please refer to the `Cúram Financial Adapter Technical Overview Guide`.

Writing-off Liabilities

All or a portion of a liability instruction can be written off so that the nominee is no longer responsible for the amount that has been written off. For example, if a person declares bankruptcy, all or a portion of that person's debt to the organization can be written off.

The amount of a write-off can be equal to or less than the outstanding amount of the liability instruction. The value of the outstanding amount on the liability will be displayed in the write-off instruction.

When a liability instruction is written off, a write-off instruction line item is created to reflect the write-off amount. This creates a record of the write-off that is useful for accountability and traceability purposes. The write-off instruction line item is automatically allocated toward the liability instruction. This reduces the outstanding amount of the liability instruction by the amount of the write-off.

Write-off instruction line items are rolled up into write-off instructions. This is true for all write-offs regardless of whether an outstanding amount remains on the liability instruction.

Note: If your organization is using an integrated environment, the liability write-off process differs from the one described here. For more information, please refer to the `Cúram Financial Adapter Technical Overview Guide`.

Over and Under Payment Processing

Case reassessment checks if changes in case circumstances may have resulted in a nominee being over or under paid. An over or under payment occurs when a new decision is created for a period that overlaps with an existing decision or decisions that have already been processed for payment or billing and that new decision differs from the existing decision or decisions. To determine the value of an over or under payment, financial data is created for the new decisions that represents what the financial components would have been if based on the changes in circumstance. This financial data is compared to the actual processed instruction line items. Case reassessment is covered in more detail in the `Cúram Integrated Case Management Guide`.

For example, John Smith received a payment in the amount of $70. This payment instruction included five benefit instruction line items, three for the child benefit component, and two for the max personal component. New evidence was recorded after this payment instruction which resulted in new decisions. The decision information overlapped with the processed financial components, and showed

different amounts that should have been paid for the child benefit and max personal components based on the changes in circumstances. The below table demonstrates the granular representation of over and under payment processing, where each 'actual amount' is a processed instruction line item and each 'reassessed amount' is financial data representing what should have been paid or billed based on the changes in circumstance.

Table 1. Granular Representation of Over and Under Payments.

This table describes the granular representation of over and under payments.

From Date	To Date	Component	Actual Amount	Reassessed Amount	Difference
16/02/2009	22/02/2009	Child Benefit	40.00	10.00	-30.00
23/02/2009	01/03/2009	Max Personal	5.00	20.00	15.00
23/02/2009	01/03/2009	Child Benefit	40.00	10.00	-30.00
02/03/2009	08/03/2009	Max Personal	5.00	20.00	15.00
02/03/2009	08/03/2009	Child Benefit	40.00	10.00	-30.00
				Total Over or Under Payment	-60.00

Processing and Maintaining Payments Received

A payment received is an amount of money that is received by the organization and recorded on the system. Payments received are used to pay off liabilities. Payments received are not generated as part of case processing. They are recorded manually by a user or through a batch process that records payments received in bulk.

Introduction

This chapter provides information on processing and maintaining payments received. A payment received is an amount of money received by the organization and recorded on the system. Payments received are usually sent to the organization in response to a bill. That is, they are used to pay off liabilities. Unlike payments and liabilities, payments received are not generated as part of case processing (the Financial Batch Suite). Instead, they are recorded on the system either manually by a user or via a batch process that records payments received in bulk, for example, all payments received by EFT from a particular bank. Payments received are associated with a person's financials through the allocation of the payment received toward one or more outstanding liabilities.

Note: If your organization is using an integrated environment, the payment received processes described in the chapter are the responsibility of the ERP financial system. For more information, please refer to the `Cúram Financial Adapter Technical Overview Guide Guide.`

Recording Payments Received

Payments can be received by the organization from persons, employers, or unknown sources. Payments can also be received in bulk through batch processing. For example, the organization may have an agreement in place with various banks to receive EFT payments in bulk, and this is managed through a batch process.

Recording Payments Received from a Person or Employer

When a payment is received, the system creates a payment received instrument, instruction, and instruction line item to reflect the information entered about the payment. The payment received instruction is added to the person or employer's list of financials. It can then be allocated toward a liability that the person or employer holds with the organization.

Recording Payments Received in a Suspense Account

When a payment is received from an unknown source, that payment is recorded in a suspense account until the organization can determine who the payment is from.

Transferring Payments Received from a Suspense Account

Payments recorded in a suspense account can be transferred to a person or employer's list of financials. A record of the payment received is maintained in the suspense account for accountability and traceability purposes. When transferred, the status of the payment received becomes "transferred" and a transfer date is recorded. After the transfer has been made, the received payment can be allocated toward a liability that the person or employer holds with the organization.

Allocating Payments Received

A payment received can be allocated toward any number of outstanding liability instructions by the person or employer from whom the payment was received. For example, if a person makes a payment of $100 to the organization, that $100 can be allocated toward one or more of that person's outstanding liabilities.

The amount of the allocation is deducted from the outstanding amount of the payment received instruction amount and applied toward the outstanding amount for the liability instruction. The allocation amount cannot be greater than the amount of the payment received. The allocation amount must also be less than or equal to the outstanding amount on the liability instruction, unless over allocation processing is set up on the liability.

To assist agency workers in allocating available funds towards outstanding liabilities, the unallocated amount on a payment received is tracked over time and displayed on the payment received instruction.

Liability Over Allocation

A liability sent out by the organization may only be an estimate of what a participant should be billed. A participant may therefore send the organization more money than the billed amount. A liability product can be set up to allow for the amount sent in to be read as the correct amount to be allocated toward liabilities. This is called liability over allocation.

For example, an employer is issued a bill for service fees in the amount of $100. That same employer sends in a payment for $120. If over allocation functionality is set up for the liability, then the full $120 can be allocated toward the liability line item. A new over allocation instruction line item is created in the amount of $20. This $20 remains as a credit reserved to the liability case. By reserving the amount, the original amount billed can be reconciled with the amount that should have been billed to determine whether the extra amount should be applied to additional liabilities on the case.

To allow this reserved amount to be applied toward additional liabilities on the case, the Reconcile Case Account batch process must be run. This batch process

starts by unreserving the credit amount. It then looks for outstanding liabilities on the case and, if found, allocates the credit amount toward them. Any remaining credit amounts are made available for allocation to other liability cases the participant may have with the organization.

If over allocation functionality is not set up for a liability, the total allocations toward the liability cannot exceed the total amount for the liability. Taking the example above, only $100 of the payment received can be allocated toward the liability instruction line item. Note that if more than one allocations are made toward a liability line item, the total of all these allocations cannot exceed the total amount for the liability ($100 in the example above).

Over allocation processing is set up for a liability product as part of financial administration.

Refunding an Unallocated Amount

All or part of the unallocated amount of a payment received can be refunded to the person or employer from whom the payment was received. The amount will be refunded through the preferred method of payment that is defined for the client. For example, a payment of $100 was received from a client and $80 is subsequently allocated towards an outstanding liability. However, if there are no further outstanding liabilities for the client, the remaining unallocated amount of $20 can be issued back to the client. If the preferred payment method for the client is check, the system will generate a check payment of $20 for the client.

When the unallocated amount has been refunded, the unallocated amount for the payment received instruction will be updated and a refund instruction line item will be created to issue the payment to the client. An allocation line item is also created to link the refund payment to the original payment received instruction.

A liability sent out by the organization may only be an estimate of what a participant should be billed. A participant may therefore send the organization more money than the billed amount. A liability product can be set up to allow for the amount sent in to be read as the correct amount to be allocated toward liabilities. This is called liability over allocation.

For example, an employer is issued a bill for service fees in the amount of $100. That same employer sends in a payment for $120. If over allocation functionality is set up for the liability, then the full $120 can be allocated toward the liability line item. A new over allocation instruction line item is created in the amount of $20. This $20 remains as a credit reserved to the liability case. By reserving the amount, the original amount billed can be reconciled with the amount that should have been billed to determine whether the extra amount should be applied to additional liabilities on the case.

To allow this reserved amount to be applied toward additional liabilities on the case, the Reconcile Case Account batch process must be run. This batch process starts by unreserving the credit amount. It then looks for outstanding liabilities on the case and, if found, allocates the credit amount toward them. Any remaining credit amounts are made available for allocation to other liability cases the participant may have with the organization.

If over allocation functionality is not set up for a liability, the total allocations toward the liability cannot exceed the total amount for the liability. Taking the example above, only $100 of the payment received can be allocated toward the

liability instruction line item. Note that if more than one allocations are made toward a liability line item, the total of all these allocations cannot exceed the total amount for the liability ($100 in the example above).

Over allocation processing is set up for a liability product as part of financial administration.

Canceling a Refund

A refund of an unallocated amount that has already been issued can be cancelled. For example, if there is a change of circumstance that results in a further liability for the client or if the check issued to the client is lost in transit it may be required that the agency cancel the refund payment.

When a refund payment is cancelled, the unallocated amount of the payment received instruction will not be updated. Instead a replacement transaction will be created that will have an unallocated amount equal to the refund payment amount that was cancelled. This is done for accountability and traceability purposes. The unallocated amount in the replacement transaction can then be allocated towards outstanding liabilities or if required it can be refunded to the client.

Canceling Payments Received

A payment received can be cancelled. For example, if an employer's check bounces after it is recorded on the system, the payment received can be cancelled to reflect the fact that the check amount was not actually received by the organization.

When a payment received is cancelled, the system creates a reversal instruction that represents the amount of the original payment received instruction.

If the payment received has been allocated toward any outstanding liabilities, these allocations must also be cancelled. For each allocation, an instruction line item is created with an amount equal to the original allocation amount. These instruction line items are liabilities that reflect the amount that had been previously cleared by the allocations, but that is again owed to the organization as a result of the payment received cancellation.

If an unallocated amount of the payment received has been refunded, these refund payments must also be cancelled. Depending upon the value of the application property curam.financial.createpaymentcorrectiononrefundcancel, the system will create either an overpayment case or a payment correction case for the amount that has already been refunded to the client. However if the refund payment issued to the client has already been cancelled, the replacement transaction that was created as a result of canceling the refund payment will instead be cancelled.

For example, a payment of $100 is received from the client and $80 is allocated towards an outstanding liability. Later, the agency refunds the remaining unallocated amount of $20 to the client through a check payment. If the payment for the $100 received from James Smith has bounced, the agency can cancel the entire payment received. When the agency cancels the payment received, any refunds associated with it must also be reversed and either an overpayment case or a payment correction case for $20 must be created to offset the refund amount that has been issued to James Smith. In the event that the refund payment of $20 has already been reversed before the cancellation of the payment received; the replacement transaction that was created as a result of canceling the refund will instead be cancelled.

Conclusion

Summary

The following list summarizes the main concepts covered in this guide:

- The financial building blocks include financial components, financial instruction line items, financial instructions, and financial instruments. These building blocks are used to process payments and liabilities, to receive payments into the system, and to perform other account maintenance tasks.
- The basic financial units are payments, liabilities, and payments received.
- Payments and liabilities are automatically generated when the Financial Batch Suite is run. Payments can also be issued online for an active case.
- Deductions and tax adjustments can be applied to benefits. Surcharge adjustments can be applied to liabilities.
- Maintenance functionality for payments includes the ability to capture manual payments, to cancel and invalidate payments, to reissue payments, and to approve suspended payments. Maintenance functionality for liabilities includes the ability to reverse and write off liabilities.
- Payments received are recorded on the system and are associated with a person or employer's financials. They can be allocated to outstanding liabilities that the person or employer holds with the organization.

Financial Instruction Types

An overview of each financial instruction type and the instruction line items relevant to that financial instruction type.

Introduction

This appendix provides an overview of each financial instruction type. It describes the potential instruction line item types that can get rolled-up into the financial instruction and any related instruction line items. It also describes, where relevant, the instruction line items that can be applied to the financial instruction types as part of set financial processing (as described throughout this guide).

Payment Instruction

The payment financial instruction type includes the benefit payments issued to eligible clients rolled up with any deductions, e.g., taxes.

Rolled-up Instruction Line Items

The following financial instruction line items can be rolled-up into a single payment instruction.

Table 2. Rolled-up Financial Instruction Line Items for Payment Instruction.

This table describes the financial instruction line items that can be rolled-up into a single payment instruction.

ILIs	Description	Created from Process
Benefit ILI, e.g., Max Personal, Child Allowance	A payment issued in relation to an eligible case component of type benefit.	Primary FC during Payment Generation
Deduction Item ILI	An amount deducted from a payment for an applied, un-applied, or third party deduction.	Secondary FC during Payment Generation

Table 2. Rolled-up Financial Instruction Line Items for Payment Instruction (continued).

This table describes the financial instruction line items that can be rolled-up into a single payment instruction.

ILIs	Description	Created from Process
Ad Hoc Bonus ILI	A once-off bonus payment that is directly related to the benefit. Ad hoc bonuses are rolled-up with the related benefit.	Ad Hoc Bonus FC during Payment Generation
Tax ILI	An amount subtracted at a set rate from a payment for tax purposes.	Process Tax Process during Payment Generation
Under Payment ILI	Changes in case circumstance can result in an eligible client being over or underpaid. Under payments can be automatically rolled-up in payment instructions thus increasing the payment instruction amount. Over payments cannot be automatically rolled-up into payment instructions. Automatic over payment processing for a benefit results in an overpayment case.	Reassessment Process

Related Instruction Line Items

The following financial instruction line items can be related to the payment instruction.

Table 3. Financial Instruction Line Items Related to Payment Instruction.

This table describes the financial instruction line items that can be related to the payment instruction.

ILIs	Description	Related via Process
Reversal ILI	When a payment instruction is cancelled, a reversal instruction line item is created to reverse the payment instruction amount. Any deductions and taxes applied to the original payment will be also be canceled. If deductions were applied to the original payment instruction, two additional reversal instruction line items are created: one that cancels the deduction amount applied to the payment instruction and the other that cancels the deduction payment instruction line item (that may or may not have been issued to a third party depending on deduction processing).	Cancel Payment Instruction Process
Manual Payment ILI	During the capture manual payment process, a manual payment instruction line item is created to record the payment made. This instruction line item is compared to the actual payment amount due, and any differences in this comparison will result in the creation of an over or under payment.	Capture Manual Payment Process

Liability Instruction

The liability financial instruction type includes the bills issued to eligible clients rolled up with any surcharges and/or over/under payments.

Rolled-up Instruction Line Items

The following financial instruction line items can be rolled-up into a single liability instruction.

Table 4. Rolled-up Financial Instruction Line Items for Liability Instruction.

This table describes the financial instruction line items that can be rolled-up into a single liability instruction.

ILIs	Description	Created From
Liability ILI, for example Service Fee	A bill issued in relation to an eligible case component of type liability.	Primary FC during Liability Generation
Surcharge ILI	An amount added to an overdue liability at a set rate.	Process Surcharge Process
Over or Under Billing ILI	Changes in case circumstance can result in an eligible client being over or under billed. Over and under billings can be automatically rolled-up in liability instructions. Over billings will decrease the amount due on a bill; under billings will increase the billing amount.	Reassessment Process

Applied Instruction Line Items

The following financial instruction line items can be applied to a liability instruction.

Table 5. Financial Instruction Line Items Applied to Liability Instruction.

This table describes the financial instruction line items that can be applied to the liability instruction.

ILIs	Description	Applied as a Result of Process
Allocated Payment Received ILI	All or part of a payment received can be applied toward the entire liability instruction or towards individual liability instruction line items.	Allocate Payment Received Process
Overallocation ILI	When overallocation is enabled on a liability, it is possible to allocate a payment received amount that is over the amount due on the liability instruction. The liability reconciliation process will reconcile the amount received against the original amount billed.	Allocate Payment Received Process
Write-Off ILI	A liability instruction can be written off in full or else all or part of individual liability instruction line items can be written off. Any written off amount will reduce the amount owed on a bill.	Write-Off Liability Process
Reversal ILI	A liability instruction can be reversed in full or else all or part of individual liability instruction line items can be reversed. Any reversed amount will reduce the amount owed on a bill.	Reverse Liability Process

Table 5. Financial Instruction Line Items Applied to Liability Instruction (continued).

This table describes the financial instruction line items that can be applied to the liability instruction.

ILIs	Description	Applied as a Result of Process
Reverse Allocation ILI	When allocations have been made towards a liability and that liability is reversed, reverse allocation instruction line items are created. These instruction line items will make the allocation amount available to be applied towards future liabilities.	Reverse Liability Process

Payment Received Instruction

The payment received instruction type is used to generate a credit financial instruction from a payment received. Payments can be received from anonymous sources, from the client, or on the client's behalf.

Rolled-up Instruction Line Items

The following financial instruction line items can be rolled-up into a single payment received instruction.

Table 6. Rolled-up Financial Instruction Line Items for Payment Received Instruction.

This table describes the financial instruction line items that can be rolled-up into a single payment received instruction.

ILIs	Description	Created From
Payment Received ILI (Unallocated Payment Received)	An amount received by the organization from a person, employer, or unknown source that has yet to be allocated toward an outstanding liability.	Receive Payment Process
Allocated Payment Received IL	All or part of a payment received that has been allocated toward an outstanding liability. Payments received can be allocated towards liabilities as part of receive payment process, thus getting rolled-up in initial payment received instruction.	Allocate Payment Received Process
Deduction Payment ILI	This is the amount equal to the third party deduction applied to a benefit which is automatically rolled-up into a payment received instruction so that it can then be allocated towards outstanding liabilities.	Secondary FC during Payment Generation
Refund ILI	All or part of the unallocated amount of the payment received that has been issued to the client. If a refund payment has been issued, it will be reflected on the payment received instruction.	Refund Unallocated Amount Process

Applied Instruction Line Items

The following financial instruction line items can be applied to a payment received instruction.

Table 7. Financial Instruction Line Items Applied to Payment Received Instruction.

This table describes the financial instruction line items that can be applied to the payment received instruction.

ILIs	Description	Applied as a Result of Process
Reverse Allocation ILI	When reversing a payment received, any allocations made towards the liability from the payment received will be checked and reverse allocation instruction line items created for each which will make the amount due again on the liability.	Reverse Allocated Payment Received Process
Reversal ILI	A payment received can be cancelled, making the funds no longer available for allocation towards outstanding liabilities. As described above, reverse allocation instruction line items will be created, if applicable.	Reverse Payment Received Instruction Process

Reversal Instructions

The reversal instruction type is used to negate all or part of an existing financial instruction. A reversal can either be a credit or a debit depending on the type of existing financial instruction being reversed. When reversing a liability instruction (or liability instruction line items) or when reversing a payment instruction through payment cancellation, the reversal is a credit. When reversing a payment received through payment received cancellation, the reversal is a debit.

Rolled-up Instruction Line Item

The following financial instruction line item is rolled-up into a single reversal instruction.

Table 8. Rolled-up Financial Instruction Line Item for Reversal Instruction.

This table describes the financial instruction line item that is rolled-up into a reversal instruction.

ILIs	Description	Created From
Reversal ILI	For reversed liabilities and payments, the reversal instruction line item is a credit. For canceled payments received, the reversal instruction line item is a debit.	Reverse Liability ILI Process OR Cancel Payment Instruction Process OR Cancel Payment Received Instruction Process

Related Instruction Line Items

The following financial instruction line items can be related to the reversal instruction.

Table 9. Financial Instruction Line Items Related to Reversal Instruction.

This table describes the financial instruction line items that can be related to the reversal instruction.

ILIs	Description	Related via Process
Benefit ILI, e.g., Max Personal, Child Allowance	When a payment is canceled, all benefit instruction line items are rolled-up into the reversal instruction.	Payment Instruction Creation Process
Additional ILIs rolled-up in original payment instruction	When a payment is canceled, all additional instruction line items rolled-up in the original payment instruction must be accounted for. These include deduction, ad hoc, tax, and under payment instruction line items.	Payment Instruction Creation Process
Liability ILI	Any or all liability instruction line items that have been reversed are rolled-up into the reversal instruction	Liability Instruction Creation Process
Over or Under Billing ILI	When a liability is reversed in full, any over or under billings rolled-up in the liability must be accounted for.	Reassessment Process
Payment Received ILI (Unallocated Payment Received)	When a payment received is canceled, all unallocated payment received instruction line items are rolled-up into the reversal instruction.	Payment Received Creation Process
Allocated Payment Received ILI	When a payment received is canceled, all allocated payment received instruction line items are rolled-up into the reversal instruction.	Allocate Payment Received Process
Reverse Allocation ILI	When allocations have been made towards a liability and that liability is reversed, reverse allocation instruction line items are created. These instruction line items will make the allocation amount available to be applied towards future liabilities. When reversing a payment received, any allocations made towards the liability from the payment received will be checked and reverse allocation instruction line items created for each which will make the amount due again on the liability.	Reverse Liability Process (when allocations have been made toward the liability) OR Reverse Allocated Payment Received Process

Applied Instruction Line Item
The following financial instruction line item can be applied to a reversal instruction.

Table 10. Financial Instruction Line Item Applied to Reversal Instruction.

This table describes the financial instruction line item that can be applied to the reversal instruction.

ILIs	Description	Applied as a Result of Process
Write-Off ILI	When all or a portion of a payment received is applied to a liability, and then that payment received is reversed, the amount applied to the liability is reversed via the creation of reverse allocation instruction line items. All or part of the reverse allocation line items can be written-off, resulting in the creation of write-off instruction line items.	Write-off Reversed Allocated Payment Received ILI Process

Write-off Instruction

A write-off instruction is a credit transaction to undo all or part of an existing liability instruction or all or part of a reversed payment received instruction.

Rolled-up Instruction Line Item

The following financial instruction line item is rolled-up into a single write-off instruction.

Table 11. Rolled-up Financial Instruction Line Item for Write-off Instruction.

This table describes the financial instruction line item that is rolled-up into a write-off instruction.

ILIs	Description	Created From
Write-Off ILI	Write-off instruction line items are created for all or part of the liability instruction, liability instruction line items, or reversed allocation payment received instruction line items and rolled-up into the write-off instruction.	Write-Off Liability ILI Process OR Write-off Reversed Allocated Payment Received ILI Process

Related Instruction Line Items

The following financial instruction line items can be related to the write-off instruction.

Table 12. Financial Instruction Line Items Related to Write-off Instruction.

This table describes the financial instruction line items that can be related to the write-off instruction.

ILIs	Description	Related via Process
Liability ILI, for example, Service Fee	The liability instruction which includes written-off instruction line item(s) is rolled-up in the write-off instruction.	Liability Instruction Creation Process

This table describes the financial instruction line items that can be related to the write-off instruction.

ILIs	Description	Related via Process
Reverse Allocation ILI	The reverse allocation instruction line items for the written-off, reversed liability or reversed payment received are rolled-up in the write-off instruction.	Reverse Liability Process (when allocations have been made toward the liability) OR Reverse Allocated Payment Received Process

Third Party Payment Instruction

A third party payment instruction is issued to a third-party in relation to third party deductions or tax deductions applied to benefits. The following financial instruction line items can be rolled-up in a third party payment instruction.

Table 13. Rolled-up Financial Instruction Line Item for Third Payment Instruction.

This table describes the financial instruction line item that is rolled-up into a third party payment instruction.

ILIs	Description	Created From
Deduction Payment ILI	The issue concern payments batch process searches for all unprocessed third party payment instruction line items for a participant and rolls them up into one payment instruction.	Issue Concern Payments Batch Process
Tax Payment ILI	This is the amount equal to the taxes applied to a benefit which can then be rolled-up into a third party payment instruction (for relevant tax authority).	Issue Concern Payments Batch Process

Adjustment Instruction

An adjustment instruction is created to correct a client's financial account. The following financial instruction line items can be rolled-up in an adjustment instruction.

Table 14. Rolled-up Financial Instruction Line Item for Adjustment Instruction.

This table describes the financial instruction line item that is rolled-up into an adjustment instruction.

ILIs	Description	Created From
Adjustment ILI	Adjustment amount to be applied to client's financial account and whether or not that amount is a credit or a debit.	Create Adjustment Process

Financial Batch Processes

An overview of each batch process in the Financial Manager application, plus a description of the prerequisite steps for running the batch processes.

Introduction

This appendix provides an overview of each batch processes in the Financial Manager application. However, before a description of each one is given, a number of prerequisites to running the batch processes, as well as some other general information on the topic, are discussed. These are as follows:

1. Submitting to the batch queue
2. Running the Batch Launcher
3. Running a batch program from the Command Line
4. Financials batch suite
5. Business processing date
6. Outputs/emails/reports

Submitting to the Batch Queue

Each batch process to be executed must first be submitted to the batch queue via the System Administrator application.

This is done by selecting the *Execute* action for that batch process from the list. If parameters are required, the user is prompted to enter them on the Execute Batch Process page. The parameters vary depending on the batch process and some are optional. Once the user enters the parameter value(s), they should then click the *Execute* button to submit the process to the queue. For batch processes that do not require any parameters, the user must confirm the execution of the process. Additionally, the user has the option of canceling at this point.

On submission to the queue, the following entities are populated, BatchParamValue only being populated if parameters are required:

- BatchProcRequest
- BatchParamValue

One important thing to note about submitting processes to the queue is that they should be submitted in the order in which they must run, so if process A must run before process B, process A must be executed first. This is particularly important when creating financials.

Running the Batch Launcher

Processes submitted to the queue are picked up by the Batch Launcher. The Batch Launcher is part of the Cúram SDE.

When the Batch Launcher is run, the first thing it does is start the stand alone keyserver - this will be required if any batch process is performing inserts onto the database.

When all the batch processes have executed, the batch launcher will stop the stand alone keyserver.

Note that batch process streaming allows multiple instances of a single batch process to be run in parallel, making full use of the available processing. Batch Process Streaming is covered in more detail in the `Batch Streaming` guide.

Running a Batch Program from the Command Line

It is sometimes useful to be able to run batch programs from the command line. The following command can be run from the main project directory, EJBServer,

substituting the appropriate values for <username>, <ClassName> and <OperationName> as well as any parameter name-value pairs:

```
build runbatch

-Dbatch.username=<username>

-Dbatch.program=curam.core.intf.<ClassName>.<OperationName>

-Dbatch.params="param1=param1value, param2=param2value"
```

Financials Batch Suite

In order for the financials records to be created, the following batch processes must be run in this sequence:

1. DetermineProductDeliveryEligibility
   - Parameters - instanceID, processingDate and productID
   - Processes - Cases to FCs
   - Pre-run Status - Approved ('CS6')
   - Post-run Status - Active ('CS1') if Eligible, PendingClosure ('CS7') if Ineligible, Suspended ('CS2') if error occurred
   - Summary of Processing - Picks up Cases with status of Approved, reassesses Cases' eligibility, and generates FCs

2. GenerateInstructionLineItems
   - Parameters - deliveryMethod, instanceID, processingDate, processingDateFrom, processingDateTo and productID
   - Processes - FCs to ILIs
   - Pre-run Status - Live ('LIV')
   - Post-run Status - Live ('LIV') if the FC is not expired; or Closed ('CLD') if nextProcessingDate > processingDate
   - Summary of Processing - Reassesses each case being processed, generates ILIs for live FCs and rolls forward their nextProcessingDate. Expires FCs if nextProcessingDate is after processing date

3. GenerateInstruments; and optionally
   - Parameters - instanceID and processingDate
   - Processes - ILIs to Instructions and Instruments
   - Pre-run Status - Unprocessed ('UNP')
   - Post-run Status - Processed ('PRO')
   - Summary of Processing - Picks up unprocessed ILIs and rolls them into instructions and instruments

4. GeneratePayslips
   - Parameters - processingDate
   - Processes - Payslips and PayslipInstructions
   - Pre-run Status - Pending Issue ('PS2') or Created ('PS1')
   - Post-run Status - Issued ('PS3')
   - Summary of Processing - Picks up Payslips pending issue and generates the Payslip for them

Business Processing Date

The Business Processing Date is an optional parameter that is common to all batch jobs. It is distinct from the system date. During batch execution the system date is always 'today' but the business processing date can be specified using the ProcessingDate parameter of the particular batch job. This gives us the ability to 'spoof' the date that a batch process was run. For example, if a financial batch run was scheduled to run on a Friday but failed. The organization could re-run it on Saturday but set the ProcessingDate to Friday's date. This means that all the calculations etc. would behave as though it was genuinely being run on the Friday. If the ProcessingDate parameter is not specified the business processing date defaults to the system date.

At a technical level, within any transactions initiated by the batch process, calls to *getCurrentDate()* will return the business processing date specified as the batch parameter. Any calls to *getSystemDate()* will return the system date, i.e. 'today', as normal.

Output Logs/Emails/Reports

When a batch process is being designed, the developer will usually want some sort of output containing summary information, such as the number of records processed, the execution time and so on. If this information is written to a log file this file can be found in the *EJBServer/buildlogs* directory.

If 'from' and 'to' email addresses were specified during the installation process then an email, containing the same information as the log file above, will be sent to the 'to' email address.

Any reports from the batch process can be found in the same location as the log and will have a .dat file extension.

GenerateInstructionLineItems

What is it?

GenerateInstructionLineItems is the batch process that produces Instruction Line Items from Financial Components.

It identifies each Financial Component of a particular delivery method that reaches its next processing date within the dates specified.

When is it run?

It is run directly after the *DetermineProductDeliveryEligibility* batch process, or at any time after a case has been activated.

Why is it run?

Its purpose is to generate Instruction Line Items that will eventually be *rolled up* by the *GenerateInstruments* batch process to produce Payment Instructions/Instruments and Liability Instructions/Instruments.

How does it work?

When the batch process is submitted to the queue from the application, it looks for the following parameters:

1. DeliveryMethod
2. InstanceID
3. ProcessingDate

4. ProcessingDateFrom
5. ProcessingDateTo
6. ProductID

Note that the parameters listed above are not mandatory. If the dates are not entered, they default to the system null date. If DeliveryMethod or ProductID is not specified, all FCs are processed regardless of the delivery method and product. The InstanceID should be specified when using the batch streaming architecture.

When the batch process executes it decides what Financial Components to process based on the parameters specified.

DeliveryMethod

If DeliveryMethod was not specified by the user, each of the following will be processed:
- Liability FCs
- Payment FCs
- Recoupment FCs
- Liability FCs, i.e., Financial Components of category 'LBY', are processed for the delivery method of Invoice.
- Payment FCs, i.e., Financial Components of category 'CLM', are processed for each delivery method listed in the MethodOfDelivery codetable. Examples include Cash ('CSH'), Check ('CHQ') and EFT ('EFT').
- Recoupment FCs, i.e., Financial Components of category 'RCP', are processed for each delivery method listed in the LibMethodOfDelivery codetable. Examples include Giro ('GIR') and Invoice ('INV').

Note that if DeliveryMethod is specified as 'INV', i.e., Invoice, only Liability FCs are processed. Otherwise, Payment and Recoupment FCs are processed for whatever DeliveryMethod was specified as, e.g., Cash ('CSH'), Check ('CHQ'), Giro ('GIR').

InstanceID

The InstanceID should be specified when using the batch streaming architecture.

ProcessingDate

This is the business date that should be used while executing the batch process. See "Business Processing Date" on page 40 details above.

ProcessingDateFrom

As mentioned above, if the user does not specify ProcessingDateFrom when submitting the batch process, it defaults to the system null date.

If ProcessingDateFrom was set (incorrectly) to a future date, it will default to the system null date when the batch process executes, otherwise, the date specified is used.

ProcessingDateTo

ProcessingDateTo is defaulted to the business processing date inside the batch process if the parameter was defaulted to a null date on submission, i.e., no date was specified by the user. The reason ProcessingDateTo is not defaulted to the current date on submission to the queue, if the user does not specify it, is because the client current date may not be 'in sync' with the server current date. In general terms, this could potentially lead to problems when the batch process executes.

If ParameterDateTo was set (incorrectly) to a future date, it will default to the business processing date when the batch process executes, otherwise, the date specified is used.

`ProductID`

If ProductID was not specified by the user, cases for all products will be processed.

Generate Instruments

What is it?
GenerateInstruments is the batch process that identifies unprocessed Instruction Line Items, i.e., those with a status of 'UNP', and processes each one accordingly.

When is it run?
GenerateInstruments is run directly after the *GenerateInstructionLineItems* batch process.

Why is it run?
It is run in order to generate the Payment Instructions/Instruments and Liability Instructions/Instruments that create the respective payments and bills that are issued to a Participant.

How does it work?
When the batch process is submitted to the queue from the application, it looks for the following parameters:
1. InstanceID
2. ProcessingDate

Note that the parameters listed above are not mandatory.

`InstanceID`

The InstanceID should be specified when using the batch streaming architecture.

`ProcessingDate`

This is the business date that should be used while executing the batch process. See "Business Processing Date" on page 40 details above.

When the batch job runs, it processes the Instruction Line Items in the following order:
- Product Delivery ILIs due, i.e., all payment and liability Instruction Line Items
- Tax ILIs due, i.e., rolls up Instruction Line Items of category 'TAX' in order to issue payments to the tax authority
- Applied Deduction ILIs due, i.e., Instruction Line Items of category 'DED' which are applied to a liability

- Repayment ILIs due, i.e., Instruction Line Items of category 'PRV'
- Third Party Deduction ILIs due, i.e., rolls up Instruction Line Items of category 'DED' in order to issue payments to third parties

GeneratePayslips

What is it?
GeneratePayslips is the batch process that identifies payslips pending issue.

When is it run?
It is run directly after the GenerateInstruments batch process.

Why is it run?
It is run in order to generate the payslips that will be sent to the Participants to reflect the breakdown of their payments (or bills). The breakdown is given at the Instruction Line Item level. As mentioned in the description of the GenerateInstruments batch process above, there are a number of different types of payslip. These are:

- Client
- Case Nominee
- Participant
- Third Party

How does it work?
When the batch process is submitted to the queue from the application, it looks for the following parameter:

- ProcessingDate

Note that this parameter is not mandatory.

```
ProcessingDate
```

This is the business date that should be used while executing the batch process. See "Business Processing Date" on page 40 details above.

When the batch job runs, it processes Payslips with a status of 'pending issue' ('PS2'). As it processes the Payslips, it decides what type of payslip it is based on its components' recipient type. It opens an output file for this payslip type and writes all payslips of this type into this file. Output data files for a particular type are only opened once for each run of the batch process.

LoadPaymentsReceived

What is it?
LoadPaymentsReceived is the batch process that loads payment received details from an external flat file onto the system for persistent storage.

What does it do?
When the batch process runs, it opens an input file and processes each record listed in the file. The data in this input file is provided in tab delimited format.

When is it run?

The batch process is run whenever the input file becomes available to the organization, e.g., from a bank, to reflect payments that have already been received. The data needs to be entered onto the system to reflect these payments.

How does it work?

When the batch process is submitted to the queue from the application, it looks for the following parameters:

1. FilePath
2. FileName
3. ProcessingDate

The first two of these parameters are mandatory.

FilePath:

The filePath will typically be of the form <Drive>:/Curam/svr/run, i.e., the directory where the input file resides.

FileName:

The fileName is the full name of the input file, including extension, that contains details of the payments received.

ProcessingDate:

This is the business date that should be used while executing the batch process. See "Business Processing Date" on page 40 details above.

When run, the batch process looks for the fileName specified in the filePath specified.

Note:

When the curam.participant.enableibanfunctionality application property is set to true then the batch input file will also require IBAN and BIC values.

IssueConcernPayments

What is it?

IssueConcernPayments is the batch process that issues payments to Persons, Employers, External Parties, Information Providers, Service Suppliers, Product Providers, Utilities, and Representatives.

What does it do?

It identifies Persons, Employers, External Parties, Information Providers, Service Suppliers, Product Providers, Utilities, and Representatives who are due to be paid and issues their respective payments. The payment method and frequency for each of the above Participant types are held on the respective entities. These are set during the registration process.

When is it run?

The times at which the process is run are dependent on the next processing dates of the Participants in question. It is conceivable that the process be run daily, as the next processing dates of the Participants, i.e., Persons, Employers, External Parties,

Information Providers, Service Suppliers, Product Providers, Utilities, and Representatives, may cover the entire week for all types. Typically, these dates will be held on the organization calendar.

How does it work?

When the batch job is submitted to the queue from the application, it looks for the following parameters:

1. ConcernTypeCode
2. MethodOfPayment
3. NextPaymentDateFrom
4. NextPaymentDateTo
5. ProcessingDate

Note that the parameters listed above are not mandatory. If the dates are not entered, they default to the system null date.

`ConcernTypeCode`

If ConcernRoleType was not specified by the user, all concern role types listed in the ConcernRoleType codetable are processed. Obviously, only Persons, Employers, External Parties, Information Providers, Service Suppliers, Product Providers, Utilities, and Representatives are of interest. Nothing will ever be picked up for the following:

- Prospect Person
- Prospect Employer

All records of the specified type will be processed even if the type does not exist, i.e., the batch process will never fail if an incorrect type has been specified. As mentioned above, payments will only ever be issued for Participants of type Person ('RL1'), Employer ('RL2'), External Party ('RL17'), Information Provider ('RL5'), Service Supplier ('RL3'), Product Provider ('RL4'), Utility ('RL6'), or Representative (''RL13').

`MethodOfPayment`

If MethodOfPayment was not specified by the user, all delivery methods listed in the MethodOfDelivery codetable are processed. Otherwise, processing is only performed for the method of payment specified.

`NextPaymentDateFrom / NextPaymentDateTo`

As mentioned above, if the user does not specify NextPaymentDateFrom or NextPaymentDateTo when submitting the batch process, they default to the system null date.

The reason NextPaymentDateFrom/NextPaymentDateTo are not defaulted to the current date on submission to the queue, if the user does not specify them, is because the client date may not be 'in sync' with the server current date. In general terms, this can potentially lead to problems when batch processes run.

Inside the batch process, two processing date parameters, `dateFrom` and `dateTo`, are set based on the values of NextPaymentDateFrom and NextPaymentDateTo. The check performed is as follows:

- If both NextPaymentDateFrom and NextPaymentDateTo are null dates, i.e., not specified by the user, `dateFrom` and `dateTo` are set to the current system date.
- If NextPaymentDateFrom was specified and NextPaymentDateTo was not, both `dateFrom` and `dateTo` are set to the NextPaymentDateFrom value.
- Similarly, if NextPaymentDateTo was specified and NextPaymentDateFrom was not, both `dateFrom` and `dateTo` are set to the NextPaymentDateTo value.
- If NextPaymentDateFrom and NextPaymentDateTo are specified, `dateFrom` is set to the NextPaymentDateFrom value with `dateTo` being set to NextPaymentDateTo.

`ProcessingDate`

This is the business date that should be used while executing the batch process. See "Business Processing Date" on page 40 details above.

Expire Payments

What is it?
ExpirePayments is the batch process that expires payments that have not being cashed after a certain length of time.

What does it do?
It identifies Payment Instruments on the system that have a reconcilStatusCode of 'issued' ('ISS') and expires them if have been on the system for a certain length of time.

Why is it run?
Its purpose is to expire payments of a specified delivery method - typically, these will be of type check ('CHQ') that have not been cashed after a certain number of days. Checks usually have a six month lifespan and cannot be cashed after this. The organization will want to run the batch process to expire the necessary payments.

How does it work?
When the batch job is submitted to the queue from the application, it looks for the following parameters:
- DeliveryMethod
- ExpiryPeriod
- ProcessingDate

Note that the ExpiryPeriod is to be entered in days, i.e., the number of days the payments have to be on the system to be picked up by the batch process. The processing date is the business date that should be used while executing the batch process. See "Business Processing Date" on page 40 details above.

ProcessPaymentInstrumentTypes

What is it?
ProcessPaymentInstrumentTypes is the batch job that processes all Payment Instrument records due for issue and writes their details into an output file.

What does it do?
It identifies the Payment Instruments that are to be issued, populates an output file with these instruments' details and updates their status to 'issued' ('ISS'). Running

this program and creating the output file is the equivalent of issuing payments. For example, if the batch process is run for DeliveryMethodType of 'EFT', the output file will be sent to the bank where the payments will be transferred to the Participant accounts.

Why is it run?

It is run in order to provide a list of payments due to the financial institution(s) that provide the payments to the Participant(s).

How does it work?

When the batch process is submitted to the queue from the application, it looks for the following parameters:.

- DeliveryMethodType
- ProcessingDate

These parameters are not mandatory. If the delivery method type is not provided by the user, all delivery method types are processed, otherwise, the one provided is processed. The processing date is the business date that should be used while executing the batch process. See "Business Processing Date" on page 40 details above.

Payment Reconciliation

What is it?

Payment Reconciliation is the batch process that reconciles an account by comparing what was due to be paid with what was actually paid, reporting any discrepancies.

What does it do?

When the batch process runs, it compares what should have been paid to a participant against what was actually paid. The input file to the process contains details of the payments received by the participants. This file comes from the institution that made the payments, for example, a bank. The payments in this flat file are compared to the payments issued by the system. Any discrepancies found are generated in a report.

Why is it run?

It is run in order to identify discrepancies, if there are any, in amounts paid against amounts that should have been paid.

How does it work?

When the batch process is submitted to the queue from the application, it looks for the following parameters:

1. FilePath
2. FileName
3. ProcessingDate

The first two of these parameters are mandatory.

FilePath:

The filePath will typically be of the form <Drive>:/Curam/svr/run, i.e., the directory where the input file resides.

FileName:

The fileName is the full name of the input file, including extension, that contains details of the payments received.

ProcessingDate:

This is the business date that should be used while executing the batch process. See "Business Processing Date" on page 40 details above.

When run, the batch process looks for the fileName specified in the filePath specified.

GeneralLedgerInterface

What is it?

GeneralLedgerInterface is the batch process that gathers financial transactions for a specified date, or date range, and exports them from the application into an output file which can then be imported into the General Ledger. The output file contains details of financial transactions at the lowest level of granularity. In the application, financial transactions at this level are defined as Instruction Line Items. The output file is a sample of what would be sent to a third party who will then use this file to update the General Ledger.

When is it run?

It is run whenever the organization wants to create a financial transaction output file that can be imported into the General Ledger.

Why is it run?

It is run in order to provide an output file of credit and debit transactions for a specified date, or date range, which can be imported into a General Ledger interface.

How does it work?

When the batch process is submitted to the queue from the application, it looks for the following three parameters:

1. CreationDateSearchInd
2. DateFrom
3. DateTo
4. ProcessingDate

Note that the parameters listed above are not mandatory. If the dates are not entered, they default to the system null date.

`CreationDateSearchInd`

CreationDateSearchInd indicates whether the extract is based on creation date or effective date. If CreationDateSearchInd was set to 'True' by the user, a creation date range search is performed, otherwise, an effective date range search is used.

When specifying the value for CreationDateRangeInd, it must be either 'True' or 'False'. It cannot be '1' or '0', or 'Yes' or 'No'. Validation is performed on the client to prevent entries of this type.

`DateFrom / DateTo`

As mentioned above, if the user does not specify DateFrom or DateTo when submitting the batch process, they default to the system null date.

The reason DateFrom/DateTo are not defaulted to the current date on submission to the queue, if the user does not specify them, is because the client current date may not be 'in sync' with the server current date. In general terms, this can potentially lead to problems when batch processes run.

Inside the batch process, the following date checks are performed:

- If DateFrom is found to be a null date, i.e., not specified by a user, an error is thrown and the batch process will not succeed.
- Similarly, if DateTo is found to be a null date, i.e., not specified by a user, an error is thrown and the batch process will not succeed.

`ProcessingDate`

This is the business date that should be used while executing the batch process. See "Business Processing Date" on page 40 details above.

ReconcileCaseAccount

What is it?
ReconcileCaseAccount is run in order to reconcile liability cases where a liability overpayment or liability underpayment has occurred. It works only for Liability Overbilling and Liability Underbilling. It does not reconcile Benefit Overpayment or Payment Corrections.

The purpose of Reconcile Case batch process is to balance a single case account. During the lifetime of a liability case, liability transactions may be added to the case. Over-allocation may have occurred to one or more of these for the purpose of reserving the money for this case therefore what the Reconcile Case batch process does is to take these over-allocations and see if they can be allocated to another liability transactions on the same case. Once all liabilities are paid, if any money remains it is made available for allocation to other cases. However, the allocation to other cases is not automated.

What does it do?
The batch process performs two separates pieces of processing:

- It reconciles all liability cases on which an underpayment has been applied.
- It also reconciles all liability cases on which an overpayment has been applied.

Why is it run?
It is run in order to reconcile both overpayment and underpayment liability cases.

How does it work?
When the batch process is submitted to the queue from the application, it looks for the following parameter:

- ProcessingDate

Note that this parameter is not mandatory.

ProcessingDate:

This is the business date that should be used while executing the batch process. See "Business Processing Date" on page 40 details above.

The process reconciles both overpayment and underpayment liability cases. It performs the processing in the following order:

1. OverPayment Liability Cases
2. UnderPayment Liability Cases

The processing performed for each type is described below.

Reconciling Overpaid Liability Cases
> This processing retrieves the list of Instruction Line Items for the Overpaid Liability case. If the instruction line items represent an over-payment and have a status of Processed, the over-allocation is converted into an unallocated payment received record. If the case the over-payment relates to has any outstanding liabilities, then the newly created payment received instruction is allocated against them.

Reconciling Underpaid Liability Cases
> This processing retrieves the list of Instruction Line Items for the Underpaid Liability case. If the instruction line items represent an under-payment and have a status of Processed, an adjustment line item record is created which will be allocated against outstanding liability instruction line items.

Financial Codetables

An overview of the financial code tables, including the code tables that you must customize to allow a product to issue financials. The ILI relationships types code table is used internally by the financial manager to process certain types of instruction line items.

Introduction

This appendix provides an overview of the financial codetables which must be customized to allow your product to issue financials. It also describes the codetable used internally to the financial manager to link related instruction line items.

Financial Codetables

The following code tables must all be customized to allow your product to issue financials. The examples below use different values for the various code table entries to help explain how they are related (e.g. "MY_RCT" and "MY_FCT"), but for simplicity you might consider using the same value for all of them (e.g."MY_COMPx").

If these code tables are not customized the TypeCode attribute of the FinancialComponent entity, and subsequently the InstructionLineItemType attribute of the InstructionLineItem entity, will be blank.

RulesComponentType
This code table lists all the component types used in the rules. You must add an entry to this code table to represent your component and reference this entry in the "name" attribute of your custom Objective Type rule class. For example:

```
<code
    default="false" java_identifier=""
status="ENABLED" value="MY_RCT"
  >
    <locale language="en" sort_order="0">
```

```
      <description>My rules component type</description>
      <annotation/>
    </locale>
  </code>
```

FinComponentType

This code table lists all of the financial component types that can be generated by the Financial Manager. You must add an entry to this code table to represent the financial schedule for your component. For example:

```
<code
    default="false" java_identifier=""
status="ENABLED" value="MY_FCT"
  >
    <locale language="en" sort_order="0">
      <description>My financial component type</description>
      <annotation/>
    </locale>
  </code>
```

ProductComponentFCConv

This code table maps the RulesComponentType codes used by the rules to the FinComponentType codes used by the Financial Manager. You must add an entry to this code table to map the custom codes for your component. The "value" should match the entry you added to the "RulesComponentType" code table. The "description" should match the entry you added to the "FinComponentType" code table. For example:

```
<code
    default="false" java_identifier=""
status="ENABLED" value="MY_RCT"
  >
    <locale language="en" sort_order="0">
      <description>MY_FCT</description>
      <annotation/>
    </locale>
  </code>
```

ILIType

This code table lists all of the instruction line item types that can be generated by the Financial Manager. You must add an entry to this code table to represent a financial transaction for your component. The "value" should match the entry you added to the FinComponentType code table. For example:

```
<code
    default="false" java_identifier=""
status="ENABLED" value="MY_FCT"
  >
    <locale language="en" sort_order="0">
      <description>My ILI type</description>
      <annotation/>
    </locale>
  </code>
```

TranslateILIType

This code table maps the ILIType codes used to represent a financial transaction to the ReassessmentAmount codes used by the eligibility and entitlement engine during reassessment. You must add an entry to this code table to map the custom codes for your financial transactions (ILIs). The "value" should match the entry from the "ILIType" code table. The "description" should match an entry from the "ReassessmentAmount" code table. Custom ILIs which are considered payments

should be mapped to the ReassessmentAmount code "AT1". Custom ILIs which are considered liabilities should be mapped to the ReassessmentAmount code "AT6". For example:

```
<code
    default="false" java_identifier=""
status="ENABLED" value="MY_FCT"
  >
    <locale language="en" sort_order="0">
      <description>AT1</description>
      <annotation>My ILI Type -&gt; Gross Amount</annotation>
    </locale>
  </code>
```

ILI Relationship Types

This codetable does not have to be customized to allow your product to issue financials, but it is used internally by the financial manager when processing certain types of instruction line items.

An example of an ILI relationship is the relationship between a deduction item ILI and the corresponding deduction payment ILI. The relationship between ILIs is held on the Instruction Line Item Relation entity.

The main data held on this entity is the unique identifiers of the related ILIs and their relationship type. A full list of all possible relationship types is given below:

Table 15. ILI Relationship Types

ILI Relationship Type	Description
REV	Reversals
SUR	Surcharges
TAD	Tax Deduction
TAP	Tax Payment
UTD	Utility Deduction
UTP	Utility Payment
RPP	Debt/Overpayment Payment
RPD	Debt/Overpayment Deduction
INT	Interest
LLB	Loan Liability
LRP	Loan Repayment
CDI	Case Deduction Item
CDP	Case Deduction Payment
CAN	Cancellation
RGN	Regeneration
ARV	Allocation Reversal

Financial Customization Points

You can extend financial functions by using the customization points for Cúram financials. You can enable customizations by adding a product type to the curam.financial.alternativeimpl.producttypes application property.

Introduction

This appendix provides an overview of the financial customization points.

Customization Application Property

The customization capability is enabled by adding a product type to the application property curam.financial.alternativeimpl.producttypes. More than one product type can be added by using a comma delimited list. The product types listed must be valid codes from the ProductType codetable.

If this property is found to contain the product type specific to the case currently being processed by the financial manager, the system will look for an alternative implementation of the relevant financial class using the additional application properties listed in the next section.

By default these application properties are all blank and the core class implementations are used.

Defining a Custom Implementation

The following application properties should contain the fully qualified name of the appropriate financial subclass:

- curam.financial.financialhook.createcancellation. Must contain a subclass of *curam.core.impl.CreateCancellation*.
- curam.financial.financialhook.casereassessment. Must contain a subclass of *curam.core.impl.CaseReassessment*.
- curam.financial.financialhook.createreversal. *Must contain a subclass of curam.core.impl.CreateReversal*.
- curam.core.hook.impl.PaymentInstructionLineItem.isPaymentProcessed: This method provides an ability to suspend the payment based on the maximum amount configured on the product. It provides a hook for customers to override the default implementation by providing logic on their implementation class and bind the implementation class to the PaymentInstructionLineItem interface using Guice.
- curam.core.impl.FinancialManagerHooks.setPaymentInstrumentEffectiveDate: This method provides the ability to change the effective date for a payment instrument. The default implementation can be overridden by binding a customer's class to the FinancialManagerHooks interface using Guice. The default implementation returns the effective date unchanged

If any of these are blank the core class implementation will be used.

Assessment Engine Hooks

In addition to the customization points described above there are further financial hooks available via the *curam.core.sl.infrastructure.assessment.impl.AssessmentEngineHooks* and *curam.core.sl.infrastructure.assessment.impl.ReassessEligibilityHook* classes. Some of the more important ones are listed in this section.

getRate

This method retrieves a rate to be used in the calculation of the deduction amount. This rate, if populated, will override the rate specified on the financial component.

haltPaymentProcessing

This method decides whether or not an Instruction Line Item should be generated from a Financial Component. It provides a hook for customers so that they can halt payments being generated on a case. This method returns false by default indicating that payment processing should continue as normal. Customized code may set the return to true in which case the payment will be halted. This method should only consider primary Financial Components and not deduction financial components.

manipulateFinancialComponents

This method provides the ability to change the list of financial components prior to generating the financials. The default implementation returns the list unchanged.

rollUpComponents

This method rolls up matching contiguous components. This is to minimize the number of identical, contiguous financial components on the system. If you have additional roll up criteria that you would like to be considered you will be able to specify it here.

getFCDatesAndValues

A method hook has been added to method *curam.core.impl.FCGenerationUtils.getFCDatesAndValues* which can be used to override the value returned by the default implementation. This method can be used to apply rounding logic to payments amounts generated in the financial component.

The hook can be utilised by using Guice to bind an implementation to interface *curam.core.sl.infrastructure.assessment.impl.FCGenerationHook* . The hook is optional, only one implementation may be provided and at the Platform layer there is no implementation for this hook.

Your implementation of the method hook must return a valid value; it must not return null. If your implementation does not need to modify the default return value, then it must return the default value which was passed in through the first parameter 'defaultResult'.

reassessEligibility

The *curam.core.sl.infrastructure.assessment.impl.ReassessEligibilityHook.reassessEligibility* hook provided, allows implementers to perform custom processing at the point of assessing eligibility, and also decide whether or not to allow processing to continue as normal, or halt it once control returns to the AssessmentEngine code.

The ReassessEligibilityHook class supports custom implementations of hook points at various stages during the assessment processing, on a per product basis. The custom implementation should be configured using the Guice dependency injection mechanism

stopFinancialProcessing

The *curam.core.sl.infrastructure.assessment.impl.ReassessEligibilityHook.stopFinancialProcessing* hook provided here, allows implementers to perform custom processing at the point of assessing eligibility, to decide whether or not to halt the processing after storing determinations and decisions and prior to performing any financial calculations.

The ReassessEligibilityHook class supports custom implementations of hook points at various stages during the assessment processing, on a per product basis. The custom implementation should be configured using the Guice dependency injection mechanism

filterNewCaseDecisions

The *curam.core.sl.infrastructure.assessment.impl.ReassessEligibilityHook.filterNewCaseDecisions* hook provided here, allows implementers to perform custom processing at the point of assessing eligibility, to decide whether or not to filter any of the new decision created by the current reassessment, before those decisions are used in the payment/decision reconciliation process (i.e. used to generate new financial components and/or over/underpayments).

The ReassessEligibilityHook class supports custom implementations of hook points at various stages during the assessment processing, on a per product basis. The custom implementation should be configured using the Guice dependency injection mechanism

listCaseDecisionsForRegenerateFinancials

The *curam.core.sl.infrastructure.assessment.impl.ReassessEligibilityHook.listCaseDecisionsForRegenerateFinanc* hook point provided here, allows the implementer to override the default list of decisions, before those decisions are used in the regenerate financial components processing.

The ReassessEligibilityHook class supports custom implementations of hook points at various stages during the assessment processing, on a per product basis. The custom implementation should be configured using the Guice dependency injection mechanism

Assessment Engine Events

Other customization points are available in the form of events via the *curam.core.sl.infrastructure.assessment.implAssessmentEngineEvent* class. Some of the more important ones are listed in this section. Listeners which implement the AssessmentEngineEvent interface can override the methods in this class to include additional custom processing.

preReassessEligibility

This method is called before the creation of financial components on initial case activation and also on the regeneration of financial components on case reassessment. This is used for both classic and CER enabled products.

postNormalReassessment

This method is called post the creation of financial components on initial case activation and also on the regeneration of financial components on case reassessment. This is used for classic products only and is not called for CER enabled products.

postReassessEligibility

This method is called post the creation of financial components on initial case activation and also on the regeneration of financial components on case reassessment. This is used for both classic and CER enabled products

postReassessFCs

This method is called for both classic and CER enabled products when a reassessment takes place, i.e. the case is reassessed on foot of a change but an over/under payment/bill may not necessarily be created.

reviewDecisionAndComponents

This method is called once per decision, as the decision is being stored but before the value for the new caseDecisionID is set. The details of the new decisions can be inspected here before they are committed.

postInsertExamineDecisions

This method is called after newly created decision details are stored. Custom code wishing to store or link old and new decisions can do so by overriding this hook and adding an alternative implentation. This event is fired once per decision.

postProcessBenefitOverpayment

This method is called after the creation of a Benefit Overpayment Case during case reassessment. Custom processing wishing to store the newly created case details can be added here.

postProcessBenefitUnderpayment

This method is called after the creation of a Benefit Underpayment Case during case reassessment. Custom processing wishing to store the newly created financial component or newly created case details, depending on which was created, can be added here.

postProcessLiabilityOverpayment

This method is called after the creation of a Liability Overbilling Case during case reassessment. Custom processing wishing to store the newly created case details can be added here.

postProcessLiabilityUnderpayment

This method is called after the creation of a Liability Underbill during case reassessment. Custom processing wishing to store the newly created financial component or newly created case details, depending on which was created, can be added here.

postProcessPaymentCorrection

This method is called after the creation of a Payment Correction Case during case reassessment. Custom processing wishing to store the newly created case details can be added here.

Notices

This information was developed for products and services offered in the United States.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM® product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing IBM Corporation North Castle Drive, MD-NC119 Armonk, NY 10504-17

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing Legal and Intellectual Property Law IBM Japan Ltd. 19-21, Nihonbash

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

Privacy Policy considerations

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings

can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

Depending upon the configurations deployed, this Software Offering may use session cookies or other similar technologies that collect each user's name, user name, password, and/or other personally identifiable information for purposes of session management, authentication, enhanced user usability, single sign-on configuration and/or other usage tracking and/or functional purposes. These cookies or other similar technologies cannot be disabled.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM's Privacy Policy at http://www.ibm.com/privacy and IBM's Online Privacy Statement at http://www.ibm.com/privacy/details the section entitled "Cookies, Web Beacons and Other Technologies" and the "IBM Software Products and Software-as-a-Service Privacy Statement" at http://www.ibm.com/software/info/product-privacy.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at " Copyright and trademark information " at http://www.ibm.com/legal/copytrade.shtml.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Java™ and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other names may be trademarks of their respective owners. Other company, product, and service names may be trademarks or service marks of others.

IBM ®

Printed in USA