IBM Cúram Social Program Management
8.0.0

Cúram Integrated Case Management Configuration Guide

Note

Before using this information and the product it supports, read the information in “Notices” on page 56

Edition

This edition applies to IBM® Cúram Social Program Management 8.0.0.

Contents

Tables

Chapter 1. Configuring Integrated Case Management

Agencies can configure case types and products, services, and referral services. The rules and evidence types that are needed to determine client eligibility and entitlement, ongoing case management options, and security are also configurable.

Administrators can manage these settings in the runtime administration and system administration applications with application properties, code tables, and rate tables.

Configuring case types

Integrated, product, screening, and assessment case types share a number of configuration options. Most integrated case type configuration options are common to other case types, and product case types have extensive configuration options.

For caseworkers to create cases at the case level, integrated case types and product case types are needed. Screening and assessment case types can be used in case management processing, and the basic configuration options that are specific to assessment and screening case types are described.

Configuring common case information

The following configuration options are common to more than one case type. For example, case display configuration options are common to both integrated case types and product case types. Most of these configuration options are also common to assessment and screening case types.

Common information includes defining a name for each case type, configuring a case home page for each case type, and important date settings.

Additionally, the agency can create associations between related types of case. For example, the agency might want to group a number of related products. To meet potential translation needs of clients, a setting enables agencies to determine whether a client's translation requirements are evaluated automatically by the system.

Important case display settings are also provided which determine where case information for the case type is displayed in the application. Agencies can use these settings to control where cases based on the case type are displayed as needed.

Defining the name of the case type

A unique and easily identifiable name must be defined for each case type. This name displays everywhere the case is referenced in the application.

When a name is defined for an integrated case type, that integrated case type automatically becomes available as a category to which products can be assigned when they are configured. For more information, see “Assigning products to categories” on page 8.

Note: A case type name is defined for all case types.

Defining the case home page

A case home page must be configured for each case type for it to be created at the case level. The case home page allows caseworkers to view case details for all cases based on this case type. The case home page setting defines the name of the `*.jsp` page that is used.

Default home pages for each case type, which can be overridden by agencies as needed.

Note: A case home page is configured for all case types.

Date settings

Date settings dictate the period during which the case details are effective. The following date settings are provided: the start date and the end date. If a start date is not specified, the system automatically sets the start date to be the current date.

If an end date is configured for a product, the end date does not prevent a caseworker from creating a product delivery case for that product type. Even if the end date is exceeded, a caseworker can create a product delivery case within an integrated case.

For some solution case types, products can be withdrawn so that new applications cannot be created. However, an administrator cannot withdraw a product just by configuring the end date. Code changes are needed to update other items, such as the screening scripts, application scripts, and eligibility rules.

Note: Date settings are common to screening, assessment, and product case types.

Grouping related case types

Use the Type setting to create associations between related types of cases. For example, to group a set of related benefit products and make them available from a drop-down list at the case level. Existing types are available for association with a case type from the type drop-down list.

Product case types are associated with the `ProductType` code table. You can dynamically add a type to this code table and publish it in system administration. Alternatively, a new type can be dynamically added when you create a product. If a new type is manually added, the system displays it in the Type drop-down list, where it can be associated with other product case types as needed. Manually added types are automatically added to the `ProductType` code table.

A case can be grouped under multiple types. For example, a product case type can belong to both a rehabilitation type and a cash assistance type.

Note: The Type setting is common to product case types and assessment case types.

Determining whether a translator is needed

The Determine Translator Required indicator dictates if a client's translation requirements are evaluated automatically by the system within each case in which the client is serving as a case member. If this indicator is set, a client's translation requirements are automatically evaluated based on their preferred language and the case owner's language skills.

If the case owner does not have a defined language skill that matches the preferred language of the client that is recorded during registration, the system automatically determines that translation services are required.

If the case owner has a defined language skill that is the same as the client's preferred language, the system automatically determines that translation services are not required. The system also determines that translation services are not required if no preferred language is recorded for the client.

For more information about defining language skills for users, see the Cúram Organization Administration Guide.

Note: The translation requirements setting is common to integrated case types and product case types.

Configuring the display of case information

A number of settings dictate where a case type is displayed in the application. These settings are common to integrated case types and product case types.

The following table describes these settings.

Table 1. Case Display Configuration Settings	
Configuration Setting	**Description**
Display in Participant Programs List	This setting dictates whether cases based on this type are displayed in the participant's list of programs. If this setting is indicated, cases based on this type are displayed in a participant's list of cases.
Display in Curam Universal Access	This setting dictates whether contact details for the case owner are displayed when a logged-in citizen accesses their account information from the IBM Cúram Universal Access online application. For case owner details to display as part of a citizen's account information, the IBM Cúram Universal Access module must be installed. For more information about Universal Access, see IBM Cúram Universal Access.
Display in My Cases Filter	This setting dictates if this case type is to be available as a filter option on the **My Cases** page. If enabled, caseworkers can filter cases that are returned on the **My Cases** page by this case type when they access their case load. By default, this setting is enabled.
Display in Case Search Filter	This setting dictates if this case type is to be available as a filter option on the **Case Search** page. If this setting is indicated, caseworkers can filter the case search by this case type. By default, this setting is enabled.

Configuring concerning participants for contact logs

Use the Members Only for Contact Log indicator to determine whether only case members are available for selection as the concerning participant of a contact created within the contact log of an integrated case, or whether all case participants are available. By default, this setting is disabled and all case participants are displayed.

Note: This setting is also available for screening case type.

Configuring a case ownership strategy

Agencies can configure a strategy for assigning initial case ownership. An initial case ownership strategy that determines the initial case owner is provided. Administrators can override this default strategy as needed.

An organization can override the default case ownership strategy as required depending upon its requirements to assign case ownership to any user, organization unit, position, or work queue. An administrator also can define a case ownership strategy on a case-by-case basis that uses workflow.

For information about assigning case ownership to users and organization groups, see the Cúram Organization Administration Guide.

Initial case ownership strategy

For integrated cases, the system automatically determines the initial case owner as follows:

1. The administrator of the participant who is the primary client of the integrated case is set as the case owner.

2. If no administrator exists for the primary client, the currently logged in user is set as the case owner.
3. If the participant administrator has no active position within the organization structure, the system assigns case ownership to the currently logged in user.
4. If the currently logged in user has no active position in the organization structure, the user receives a validation message that states that the case cannot be created because no case owner can be identified.

For product delivery cases, the system automatically determines the initial case owner as follows:

1. The case owner of the integrated case to which the product delivery belongs is initially set as the case owner.
2. If no related integrated case exists, the administrator of the participant who is the primary client of the product delivery case is set as the case owner.
3. If no administrator exists for the primary client, the currently logged in user is set as the case owner.
4. If the user who registered the primary client has no active position within the organization structure, case ownership is assigned to the currently logged in user.
5. If the user who is the case owner of the related case has no active position within the organization structure, the supervisor of the related case is assigned case ownership.
6. If the supervisor has no active position or if no supervisor is assigned to the related case, case ownership is assigned to the organization unit to which the case owner of the related case belongs.

Assigning case ownership by using workflow

The Ownership Strategy setting allows administrators to configure a case ownership strategy for each case type by using workflow. This setting defines how the initial case owner for the case is determined. If an ownership strategy is defined for a case type, this setting defines how the initial case owner for the case is determined. For example, an administrator can specify that a particular type of integrated case is assigned to a specific work queue.

When a case ownership strategy is defined for a case type, the system transitionally sets the initial owner of a newly created case to the `Temporary Owner Assignment` work queue while the workflow process determines the initial case owner. When the workflow process completes, the case is reassigned to the case owner who is determined by the workflow process. Two system application properties relate to this work queue.

- The `curam.workflow.logtempownerworkqueueassignment` application property is used to dictate whether case transaction log events that relate to the temporary owner assignment are automatically recorded by the system. The default value is `NO`. If this value is set to `YES`, a `User Role Canceled` case transaction log event is automatically recorded when the case is reassigned from the `Temporary Owner Assignment` work queue.
- The `curam.workflow.displayworkqueueasownerincaseuserrole` application property is used to dictate whether the temporary assignment work queue displays as a case owner on the **User Roles** page in a case. The default value is `NO`.

  Important: When administrators assign case ownership to a workflow, they must ensure that the case ownership can be resolved through the workflow process that is associated with the case type. Otherwise, the case creation process fails.

Configuring assessment case types

Assessment functionality is provided by the Cúram Decision Assist and IBM Cúram Outcome Management modules. Administrators can maintain assessment case type configurations, but new assessment case types cannot be added in the application administration.

For more information about configuring assessments, see the Cúram Decision Assist overview and Configuring Outcome Management.

Assessment case type maintenance includes the maintenance of common assessment information, such as home page configuration, described in “Configuring common case information” on page 1.

Assessment configuration also includes the assignment of eligibility rules to an assessment. This rule set is applied to evidence to determine potential eligibility for products. Rule sets implemented in Cúram Rules are assigned to an assessment case type. The process begins with selecting the rule set. The rule set is searched for and selected from the list of existing rule sets.

At least one rule set must be assigned to an assessment as a prerequisite to eligibility determination. Information that is recorded for the rule set assignment includes the name of the eligibility rule set to be assigned to the assessment and the period during which the rule set is assigned to the assessment. This period cannot overlap with the periods of existing assessment rule set assignments. One or more rule sets can be assigned to an assessment. However, only one rule set can be active during a period.

Rule sets implemented in Cúram Rules are maintained by using the Cúram Rules Editor.

Configuring screening case types

Agencies can configure a screening case type and assign one or more assessments to the screening case type.

For more information about options to configure a screening case type, see "Configuring common case information" on page 1.

At least one assessment must be assigned to a screening case type for a screening to be created. You must specify the period during which the assessment is assigned to the screening case type. This period cannot overlap with the periods of existing assessment assignments.

Configuring products

Products are configured to create and deliver product delivery cases to address the needs of clients. A product can be a benefit that an agency provides to clients, a liability that the agency bills to clients or businesses, or a payment correction.

Benefits are used to provide subsidies to eligible participants in the form of monetary payments. For example, a cash assistance product provides monetary payments to families in need. Liabilities are used to bill and receive payments from liable participants. Administrators can configure payment corrections products to correct overpayments and underpayments that are issued to clients.

Benefit and liability product configuration is extensive, so administrators can use a product wizard to create benefit and liability products. Each step in the wizard represents an important category of information that must be configured for each product. A number of configuration options can be set for both benefit and liability products, some specific settings that apply to benefit products only and liability products only.

After products are configured, rules are assigned to them to determine eligibility and entitlement for the cases they govern. For more information about assigning rules to products, see "Assigning Cúram Express Rules to products" on page 27.

Configuring benefit and liability products

Products are configured in order for users to create and deliver product delivery cases to address the needs of clients. A product can be a benefit that an agency provides to clients, a liability that the agency bills to clients or businesses, or a payment correction that is used to correct overpayments and underpayments that were issued to clients.

Benefits are used to provide subsidies to eligible participants in the form of monetary payments. For example, a cash assistance product provides monetary payments to families in need. Liabilities are used to bill and receive payments from liable participants. In addition to liability and benefit products, payment corrections are also configured as products as part of the application administration.

A dynamic product wizard is included that allows administrators to easily configure benefit and liability products. Each step in the wizard represents an important category of information that is configured for each product. A number of configuration options are common to both benefit and liability products.

Configuring general product details

General configuration options are available for each benefit and liability product. These options include settings that are applicable to multiple case types such as the case name, home page, and display settings.

For more information about common case settings, see “Configuring common case information” on page 1. The following configuration options are specific to products.

The participant types that are eligible for cases based on the product can be defined. Eligible participant types that might be associated with products include person, employer, utility; information provider; service supplier; and product provider, see “Indicating eligible participant types” on page 8.

Product categories are used to group similar products. Examples of product categories include assistance, illness, and financial. These categories are used to define the subset of products that might be added to a given integrated case, see “Assigning products to categories” on page 8.

Case reviews can be set to occur at a defined frequency, for example, every 6 months. All cases based on the product inherit the product's case review frequencies.

Setting a review frequency

Case reviews can be manually created within a case, or automatically created by the system.

The 'Review Frequency' setting is used to determine the frequency for cases based on this product. For example, a review frequency of the 15th day of every month can be configured for a product. If a review frequency is specified, when a case based on that product is approved, the system automatically inserts multiple case review events and associated caseworker tasks between the case start date and the expected end date based on the frequency configured.

Indicating whether supplier returns are accepted

The Supplier Returns Accepted setting indicates whether service supplier returns might be created for a product. A service supplier provides services or tasks that must be done by a qualified individual or body, for example, eye examinations, court translations. A service supplier return is a return that is filed by a service supplier in respect of the service that indicates the number of services that are provided for the agency.

Service supplier returns allow the agency to calculate the amount of money that is owed to a service supplier in return for services provided. For example, a registered doctor makes a return to the agency for the number of persons for whom the doctor provided eye examinations. Each individual service supplied by the doctor, that is, each eye examination, appears as a line item in the overall return.

Configuring eligibility determination and decisions

Agencies can control aspects of eligibility determination processing. Eligibility determination settings have a large impact on the determination of a client's eligibility for product deliveries and on successfully providing them to eligible clients.

Administrators can configure eligibility determination processing including how eligibility decisions display for cases that are based on this product.

Applying system-wide changes to cases

Agencies often need to change how eligibility and entitlement are determined. For example, if a rate changes as a result of a change in legislation, an administrator must change the rate in the appropriate rate table. After this change is published, it is accounted for when case reassessment next occurs within any product delivery case based on a product that uses the rate. For example, when reassessment is initiated manually within the case or when financials are generated.

Because these types of system-wide changes can affect many cases, the application can identify and reassess all cases that are affected by the change. Upon publication of these types of change, the system

captures information about the changes that occurred. For example, changes can be to a rule set, rate table, data configuration, or product configuration, such as a change to a product period or to the rule set associated with a product period. A series of batch processes automatically identify and reassess all cases that have a dependency on any changes.

If the result of reassessment differs from the current determination result for the case, then a new determination is created. For products that are configured to not allow the reassessment of closed cases, no reassessment occurs if the case is closed.

If reassessment already is initiated within an individual case, then that case is not reassessed during run of the batch processes.

Allowing open-ended cases

To allow for situations where a client might be receiving a benefit for an unknown period, you can determine whether case eligibility decisions can be open-ended with no effective end date. For example, an administrator can specify that a product for a pension benefit case allows for the case to be open ended. Financials issued in respect of the case are then paid indefinitely until circumstances change on the case, or an expected end date is set on the case. Alternatively, an administrator can specify that a different product such as one governing a dependent child benefit case is not open-ended because the child benefit is only to be paid up to a certain age of the child. In this situation, an expected end date for the case must be explicitly specified. When the child reaches that age, financials that are issued in respect of the child benefit are automatically stopped.

The Allow Open Ended Cases setting dictates if an expected end date must be defined for cases that are based on this product. If this setting is specified, an expected end date is not mandatory for the case and a user can manually enter or modify the expected end date. If no expected end date is specified, the latest eligibility decision within a determination that is created for the case will be open ended with no effective end date.

If this setting is not specified, the system automatically sets the expected end date to be the case start date upon initial creation of the case and the expected end date is automatically updated as certification information is entered on the case.

Configuring a determination comparison strategy

This setting specifies how the system decides whether a new determination result is different from the previous determination result. A determination comparison strategy must be configured for each product. A determination result can be considered different when any user-facing information changes, such as data that impacts eligibility and entitlement, data that can result in new key decision factors, or data that can result in new decision details display information. Alternatively, a determination can be considered different only when information changes affect eligibility and entitlement.

The Determination Comparison Strategy setting is used during eligibility processing for products. When case reassessment is initiated, a determination is only stored if it is different than the previous determination.

Note: The key decision factors feature is deprecated. For more information, see Deprecated features.

The determination comparison strategy values available are associated with the `DeterminationCompStrategy` code table. A new value can be added to this code table and published as part of system administration.

Configuring a decision summary display strategy

The Decision Summary Display Strategy setting determines the summary information that is displayed for each decision within a determination. For example, for eligibility decisions that include information about the client's entitlement amount, this setting can be used to determine the frequency at which this amount is displayed. Even if a client's entitlement is calculated to be $70 weekly, the client's entitlement amount can be displayed in a daily or monthly format instead.

The available decision summary display strategy values are associated with the `DetIntSummarizerStrategy` code table. A new value can be added to this code table and published as part of system administration.

Configuring a reassessment strategy

The Reassessment Strategy setting determines whether reassessment can occur in closed cases. By default, the setting is set to not reassess closed cases. For a change in circumstance within an individual case, for example a change in evidence or a change in certification, the system does not reassess the case if it is closed.

When a system-wide change such as a change to a rate table or a change to a rule set is made that might affect multiple cases, this setting is also used to determine whether closed cases are reassessed when batch processes are run to reassess all cases affected by the change. The reassessment strategy values available are associated with the `ProductReassessmentStrat` code table.

Indicating eligible participant types

Administrators can specify the participant types that are eligible for cases based on a product. Typically, only person participants are eligible for benefit products so only the Person participant type is implemented fully by default.

When this setting is enabled, product delivery cases that are based on this product type can be created for person participants so that their eligibility for the agency's products can be determined.

The following participant types can also be associated with products: Person, Employer, Utility, Information Provider, Service Supplier, and Product Provider. However, selecting any of the other eligible participant settings has no impact on application processing in a default application installation. If agencies want to extend the participant types that are eligible to receive a product to include others, such as employers or service suppliers, development effort is needed to configure the application processing that implements these settings. For more information about participants, see the Cúram Participant Guide.

Assigning products to categories

Products must be assigned to at least one integrated case category in order for cases based on that product to be created from within an integrated case. These categories are used to group similar products. Examples of categories include social assistance, illness, and financial.

Each category is used to define the subset of products that might be added to that particular type of integrated case. Products can be assigned to existing integrated case categories. Only products that belong to the selected integrated case category might then be added to that integrated case at the case level. For example, if an integrated case is assigned a category of social assistance, then only products that belong to the social assistance category can be added to that integrated case. Products can belong to multiple categories, which enables cases that are based on that product to be added to multiple types of integrated case.

Configuring product provisions and assigning them to locations

The application allows agencies to configure product provisions and associate those provisions with locations. An example of a product provision is a benefit that is offered to eligible persons on behalf of the agency.

The agency itself is considered a product provider if it provides products directly to its clients. Product providers are selected from product provider participants that are registered on the system as part of the **Participant Manager**.

Product provision settings include the start dates and end dates of the period during which the product provision is active, the estimated cost, a payment method (for example, cash), a payment frequency (for example, the first day of every month).

Product provisions can be delivered from one or more locations to suit the needs of the different eligible persons. For example, a training provision can be offered at different training center locations. Information that is recorded for the product provision locations includes the cost of the provision at the location and the period during which the provision is offered at the location. A case's product provision location can be changed dynamically to any of the provision locations stored for that case's parent product.

Defining product provisions

Product provisions are the product providers that deliver the actual products to case recipients. Product provision details must be associated with each product on creation. Product provision definition involves recording the details of the provider who provides the product. The Product Provider setting allows an administrator to search for a product provider from the product providers that are registered as part of Participant Management.

The product provider can be the agency itself or a different product provider.

Start Date and End Date settings dictate the period during which the product provision is effective. By default, the system automatically sets the start date to be the current date.

The Currency setting allows the agency to select the preferred currency for payment for the provision. The currencies that are available for selection are associated with the Currency code table. A new value can be added to this code table and published as part of system administration.

The Estimated Cost setting allows the agency to estimate the amount that the agency spends on a particular product provision. That is, the cost of a product that is offered by a particular provider. The default value is `0`. An administrator can change this value as needed.

The Payment setting determines the preferred method of payment for the provision, for example, cash. The default value is 'Cash'.

The Payment Frequency setting dictates the payment frequency for the provision, for example, the first day of every month.

A product cannot be deleted if it has an active product provision. Similarly, a product provision location cannot be deleted if active cases are assigned to that location.

Associating a location with a product provision

Each product provision has an associated location. The Location setting is used to search for the location from which the product provision is delivered. Administrators can select a location from existing locations recorded for the organization.

Start Date and End Date settings dictate the period during which the product is offered by a product provider at the location. By default, the system automatically sets the start date to be the current date.

The Cost setting is used to specify the cost of the provision at the location. The default value is `0`. An administrator can change this value as required.

Assigning a delivery pattern or billing pattern

A delivery pattern or a billing pattern must be assigned to each product on creation. These delivery patterns are available for selection at the case level when a nominee's preferred delivery pattern is assigned for a particular case component. Delivery patterns define the default frequency and payment method by which financials are issued to a nominee on cases based on this product, for example, weekly by check.

Each product can be assigned a default delivery pattern that is assigned automatically to the nominee when a case is first created. The default delivery pattern that is assigned to the product is provided as guidance, but does not have to be the preferred delivery pattern of the nominee. The following delivery pattern information can be defined for each product delivery pattern: maximum amount, default pattern, cover pattern, delivery method, and frequency.

- Delivery patterns are defined for benefit products. A delivery pattern defines how and when a benefit product is delivered to a client in the form of a payment.

- Billing patterns are defined for liability products. A billing pattern defines how and when a liability product is delivered to a client in the form of a bill.

All delivery and billing patterns include a delivery or billing method and a delivery or billing frequency. For example, a product is delivered to a client every two weeks on a Monday by EFT.

Administrator can set up new delivery patterns and billing patterns in administration. The following delivery and billing pattern information can be defined.

Defining an effective from date

The date from which the delivery or billing pattern is effective, by default, the current date.

Defining the maximum payment amount

The maximum payment amount sets the maximum amount for a payment that is issued by the delivery pattern.

Maximum payment amount is used for benefit products only, there is no limit on bills issued. If the total value of a payment instruction is greater than the maximum amount, the payment is suspended. The default value is $0.00.

The payment remains suspended and is not issued until a user manually approves the payment. User access to the approve payment function is controlled by using security settings. Users who have been assigned security access to the **approve payment** function are allowed to approve the suspended payment. Therefore, if approval of suspended payments needs to be restricted to a subset of users, security access to the **approve payment** function must be assigned to only that subset of users. For more information, see *User Security Profiles*.

Defining the default pattern

The Default Pattern setting allows an administrator to set this delivery or billing pattern as the default delivery or billing pattern for the product. If a product delivery case is created automatically, the default product delivery or billing pattern is automatically assigned to the case nominees. For example, automatic overpayment case creation assigns the default delivery pattern for the overpayment product to the case nominees. When a product delivery case is created manually, this default can be overridden on a case-by-case basis.

Setting the cover pattern

The Cover Pattern setting defines the cover period for the delivery of the payment or bill. A cover period dictates how payments or bills are issued, such as in advance, in arrears, or once-off. For example, the delivery pattern, weekly by check on Mondays in advance, indicates that each payment is made on a Monday. The payment covers the week that starts on the Monday and continues to the next Sunday.

The available cover patterns are associated with the `ProductCoverPeriod` code table. The default cover pattern is Issue in Advance. A new value can be added to this code table and published as part of system administration.

Defining the delivery method

A delivery method is a method by which financial transactions can be issued or received for a product. Delivery methods can apply to benefit products only, to liability products only, or to both. For example, the invoice delivery method is only applicable to liability products and can be configured to apply to liability products only. When administrators create a new product, they can select an existing delivery or billing method from the drop-down list of existing methods that are configured. This setting is mandatory. Administrators can set up new delivery methods. Each newly created delivery method is automatically available for selection when a new product is created.

Defining an offset for the delivery method

Each delivery or billing method can have an offset. The offset is the number of days before the due date by which a financial component must be processed so that the benefit or liability reaches the nominee on time. For example, delivery and billing methods for benefits include cash, check, EFT, and vouchers. The check delivery method can be configured with an offset of three days in advance to allow for bank clearing. This offset ensures that funds are available to the participant on the due date. The default offset for a delivery method is '0'.

Defining the delivery pattern frequency

The Delivery Pattern Frequency setting dictates the frequency by which the payments or bills for a product are issued to eligible participants. For example, a delivery pattern might be 'Recur every one week on Monday' or 'Day 1 of every 1 month'. This setting is mandatory.

Configuring case reassessment and adjustments financial processing

Agencies can control aspects of case reassessment processing and how adjustments to payments are made. Case reassessment settings determine whether overpayment cases, underpayment cases, or payment correction cases are created automatically when an overpayment or underpayment is discovered on reassessment. Adjustment settings allow agencies to correct a client's financial account, so that the system does not attempt to generate a compensating overpayment or underpayment.

For example, adjustment functionality allows the agency to apply appropriate taxes to payments issued in respect of a benefit product.

When adjustment is required for liabilities, if an agency is billed monthly for a service, the agency has one month to send payments toward the bill. If the agency does not send in payments within the month, surcharges are applied to the outstanding bill.

These configuration options affect financial processing for cases based on a product. The options include adjustment settings and case reassessment settings. Case reassessment is used to re-evaluate payments or bills when there is a change in circumstance on the product delivery case. Examples of changes in circumstance include changes in evidence or certification.

Reassessment and financial processing are complex. For more information about reassessment and over and underpayments, see Developing with Eligibility and Entitlement by using Cúram Express Rules. For information about financial processing, see the Cúram Financials guide.

Indicating whether adjustment is required for a product

The Adjustment Required setting indicates whether adjustments are required for a product. If this 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. If the Adjustment Required setting is turned on for a liability product, surcharges will be automatically 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 for the product. The adjustment rate for surcharges is set at a fixed rate. For example, the agency may specify that liabilities left unpaid for one month will be surcharged at five percent.

Indicating the adjustment frequency

The Adjustment Frequency setting is used to dictate the timeframe after which surcharges are applied to any outstanding liabilities that have not been paid. For example, if an agency is billed monthly for a service, and the adjustment frequency of that service is set to one month, then the agency will have one month to send in payments toward the bill. If the agency does not send in payments within the month, surcharges will be applied to the outstanding bill.

Configuring underpayment case creation

If an underpayment is discovered during reassessment, a new case for the underpayment can be created to resolve the underpayment amount. This case can either be an underpayment case or a payment correction case. The Automatic Underpayment Case Creation setting is used to indicate whether an underpayment case is automatically created when an underpayment is discovered by the system on reassessment. This setting works along with the Use Rolled Up Reassessments setting, see “Configuring benefit products” on page 14.

For more information about payment corrections, see “Configuring payment correction products” on page 17.

If underpayment or payment correction case creation is not set to be automatic, an underpayment financial component is created on the original benefit or liability case. For benefits, both the original benefit case and underpayment or payment correction case can be used. For liabilities, only the underpayment case type can be used to pay a case recipient. Underpayments for liabilities are used to pay case recipients when they have been over charged. This financial component is once-off in the amount of the underpayment.

Additional functionality is provided for benefit underpayment processing. Benefit underpayment processing can be configured so that underpayment cases are only created for a case recipient with outstanding liabilities. To achieve this, an application property is available within system administration. Use the `curam.miscapp.checkforliveliabilities` application property to support the allocation of an underpayment toward an outstanding liability. This property determines whether a check is performed by the system to establish the existence of live liabilities for a client before an underpayment case is created for that client. The default value of this property is YES.

Configuring overpayment case processing

If an overpayment is discovered during reassessment, an overpayment case or payment correction case can be created to resolve the overpayment amount. For benefits, both an overpayment case and payment correction case can be used to bill a case recipient for the amount they were overpaid. For liabilities, only the overpayment case type can be used to bill a case recipient. Overpayments for liabilities are to bill case recipients when they have not been billed sufficiently. For more information about payment corrections, see “Configuring payment correction products” on page 17.

The Overpayment Case Processing setting dictates how the system manages the discovery of an overpayment during reassessment. For benefits, the setting works along with the Use Rolled Up Reassessments setting described in “Configuring benefit products” on page 14.

The Overpayment Case Processing setting can be configured in one of the following ways:

- Administrators can specify that a case is created automatically when an overpayment is discovered during case reassessment to resolve the overpayment. The case can be an overpayment case or a payment correction case depending on the value that is specified for the Use Rolled Up Reassessments setting. After the case is created, users must manually approve, activate, and generate the liability financials that are required to correct the overpayment.
- Administrators can specify that a case is automatically created and approved, activated, and liability financials generated without the intervention of a caseworker. This option is available only for benefit products for which the Use Rolled Up Reassessments setting is set to 'No'.
- The system does not automatically create a case to correct the overpayment. Instead, a task is generated to alert the caseworker of the overpayment. The caseworker can then manually create and manage an overpayment case to correct the overpayment.

When a benefit product is initially created, the default value for this setting is 'Create Overpayment Case'. This value can be changed by an administrator. For liability products, the Overpayment Case Processing setting can be set to 'Yes' or 'No'. Yes indicates that an overpayment case is created automatically when an overpayment is discovered during case reassessment to resolve the overpayment. After the case is created, users must manually approve, activate, and generate the liability financials that are required to correct the overpayment. A setting of 'No' instructs the system not to automatically create a case to correct the overpayment. Instead, a task is generated to alert the caseworker to the overpayment. The

caseworker can then manually create and manage an overpayment case to correct the overpayment. When a liability product is initially created, the value for this setting is defaulted to 'No'. This value can be changed by an administrator.

Date list and rerate frequency

Two case reassessment settings apply only to products that are configured for Cúram rules.

The `Date List` setting determines the type of date list that is used by the assessment engine to create case decisions. Date lists include the event date list and the pattern date list:

- The event date list is a list of dates that are built into a product when the product is created. These dates include core dates that are implemented in a default application installation, such as the case creation date, the certification start date, the date after the certification end date, and the effective date of the product rule set. They also include any custom dates that are specific to individual products. These custom dates must be added to the list of the product's significant dates by a developer during the initial design of a product.
- The pattern date list is a list of dates that is dynamically created by the assessment engine. The assessment engine creates this date list based on the rerate frequency for the product and the delivery pattern for the case nominee.

The `Re-rate Frequency` setting is used by the assessment engine to compile the list of dates on which the rules engine is called to create case decisions. When it uses a pattern date list, the assessment engine compares each case decision to the case decision that follows it. If a difference is found between these two decisions, the assessment engine examines every date between the two decisions to determine the exact date when case circumstances change.

Configuring certification requirements

Certification requirement settings are used to indicate whether the product requires some form of certification for eligibility determination. Certification settings include a setting to specify whether certification is required, the frequency at which a participant provides certification, and the grace period to allow a participant to retain eligibility after certification expires.

Indicating whether certification is required

Certification is the process of certifying that a participant is eligible to receive a benefit. For example, a disability benefit product might require that a doctor certifies a participant's inability to work once a year to ensure that they still meet the eligibility criteria. The Certification Required setting can be used to indicate whether the product requires some form of certification for eligibility determination. For products that are configured to be non-certifiable, no certifications can be recorded within cases based on the product. If the Certification Required setting is enabled, a Certification Frequency must be specified.

This setting is not directly used in any eligibility determination processing by default. You might use it, for example, in a customized rule set to determine a client's eligibility.

Setting a certification frequency

The Certification Frequency setting indicates the frequency at which a participant is expected to provide certification to verify eligibility. The system automatically compares case certifications to the certification frequency and displays an informational message when the certification period does not align with the certification frequency. For example, if the certification frequency is set to `Day 1 of every 1 month(s)` and the certification period that is entered covers only a two-week period, an informational message is displayed to alert the caseworker that the period of the certification is different from the certification frequency.

Setting a certification grace period

The Certification Grace Period setting can be used to allow a participant to retain eligibility for a specified number of days after certification expires.

This setting is not directly used in any eligibility determination processing by default. You might use it, for example, in a customized rule set to determine a client's eligibility.

Product access security settings

Agencies can use product access security to restrict users' ability to read, create, modify, or approve cases that are based on the product. User security profiles are defined by a hierarchy of security elements called security identifiers (SIDs). These SIDs are the building blocks of application security.

Products can be associated with specific SIDs. In order for a user to access a particular product, that user's security profile must contain the SID that is associated with that product. For example, if a product is associated with an SID, then a user cannot access a case based on that product unless that SID is in their security profile. If no SID is entered for a product, then all users have access to that product. However, if at least one SID is entered for a product, then users without that SID in their profile cannot access that product.

The following list describes the rights that can be secured for each product:

Approve
Any user whose security role contains an approve SID has the security privileges to approve (or reject), read, and maintain cases based on this product.

Create
Any user whose security role contains a create SID has the security privileges to create and read case information for cases based on this product.

Maintain
Any user whose security role contains a maintain SID has the security privileges to maintain and read case information for cases based on this product. Maintain rights for product security includes rights to manage case evidence, manage case certification, and check case eligibility.

Read
Any user whose security role contains a read SID has the security privileges to read case information for cases based on this product. For product security, a user with read rights can view case details. Any users with maintain, create, or approve rights can also read case details. When you set up product-based security, the read rights are there to be assigned to users who can only view case details.

Location-based security

Location-based security can also be used to restrict a user's case and client access based on a combination of the location security for the organization and the product access security for the product. Security can be applied on a location only basis or by using a combination of location and product security settings. A user can access cases only in their location or sublocations. However, if product access security is set up for the product that governs a case, the user might be further restricted when they try to access the case. For example, if a user does not have the relevant product access security to view a case as part of their security profile, they are denied access to the case. For more information about location-based security, see the Cúram Location Administration Guide.

Configuring benefit products

Settings specific to benefit products are used to specify how over and underpayments are created upon reassessment and how deductions are made from cases that are based on that benefit product. Deduction-specific settings apply to third-party deductions (deductions that are made from a participant's benefits to make payments to a third party such as a utility company), applied deductions (deductions that are applied to a participant’s outstanding liabilities, and unapplied deductions (any other deduction from a participant's benefit payment). These settings allow limits to be set on the total deduction amount that can be deducted from payments that are issued for the benefit product.

Rolled up reassessment

The Use Rolled Up Reassessments setting is used to determine whether over and underpayment amounts are broken down and displayed by case component for a nominee or rolled up into one amount that combines the amounts over and underpaid for each case component. If the Use Rolled Up Reassessments setting is enabled, upon case reassessment, either an overpayment case or an underpayment case is created. The overpayment case or underpayment case contains one benefit overpayment or benefit underpayment component for the rolled up amount. For example, if reassessment occurs due to a change in circumstance that results in an overpayment of an income support component in the amount of $50 and an underpayment of a medical allowance component in the amount of $20, then an overpayment case is created to represent an overpayment of $30. When the overpayment liability is created within the payment correction case, one liability line item is created and displayed for the consolidated overpayment amount of $30.

If this setting is not enabled, a payment correction case is created upon case reassessment. A payment correction case can represent either an underpayment or an overpayment and allows users to see over and underpayment amounts that are broken down by case component for a nominee. Whether the payment correction represents an underpayment or an overpayment is dictated by whether the total balance of the payment correction case amount is an overpayment or underpayment. For example, if reassessment occurs due to a change in circumstance that results in an overpayment of an income support component in the amount of $50 and an underpayment of a medical allowance component in the amount of $20, then a payment correction case is created to represent an overpayment of $30. When the overpayment liability is created within the payment correction case, two liability line items are created and displayed, one for an overpayment of $50 for the income support component and one for an underpayment of $30 for the medical allowance component. For more information about payment corrections, see “Configuring payment correction products” on page 17.

Maximum deduction rate

The Maximum Deduction Rate setting is used to determine the maximum percentage of a benefit payment that can be deducted from the total payment amount. For example, if the maximum deduction rate is set to 50 for a benefit product, then the total amount of all deductions for a case based on that product cannot exceed 50% of the benefit payment amount. Deductions are processed in the order that they are recorded on the system. The first deduction that causes the deduction percentage to go over the maximum deduction rate for the case will prevent any future deductions from being processed.

Minimum deduction amount

The Minimum Deduction Amount setting is used to determine the minimum amount of money that can be deducted from cases based on this product. Deductions are not processed if they fall under this minimum amount. For example, if the minimum deduction amount is $10, then a deduction cannot be processed unless it is greater than $10. The minimum deduction amount is intended to reduce the number of small payments that are received by utilities and other companies, who might prefer to receive multiple deductions rolled up into one payment to simplify financial processing.

Minimum payment amount

The Minimum Payment Amount setting is used to determine the minimum amount of money that a participant must receive after all deductions are processed against the total benefit payment. For example, if the minimum payment amount is $40 and a deduction is scheduled that causes the total benefit payment to fall to $30, the deduction is not processed.

Deduction application properties

Two system administration properties relate to the creation of deductions at the case level:

- Use the `curam.case.deduction.appliedDeductionParticipants` application property to dictate which participants can be searched when a caseworker searches for a participant whose liability an applied deduction will be allocated toward. The default value is Employer, External Party, Person,

and Utility meaning that employers, external parties, persons and utilities can be searched when creating an applied deduction at the case level.

- Use the `curam.case.deduction.thirdPartyDeductionParticipants` application property to dictate which participants can be searched when a caseworker searches for a third-party participant who will receive a deduction amount. The default value is Employer, External Party, Person, and Utility meaning that employers, external parties, persons, and utilities can be searched across when creating a third-party deduction at the case level.

Associating deductions with benefit products

In addition to the deduction settings application properties, both new and existing deduction types that exist on the system can be associated with a product. This association allows deductions to be created at the case level for any cases that are based on that product. In addition to configuring details about the deduction type, each deduction can be assigned a priority that dictates the order in which it is processed when payments are generated for cases that are based on the product.

If a deduction is already associated with the product and has the same priority as the new or updated deduction being associated to the product, the priority of the existing deduction will be automatically updated to have the next priority. The priority of any other existing deductions can also be impacted as a result. A sequencing function automatically increases or decreases the priorities of any other deductions that are associated with the product. For example, if a deduction is added with a priority of 1, and a deduction exists with a priority of 1, the new deduction is stored with a priority of 1 and the existing deduction is stored with a priority of 2.

Configuration is also available that provides the ability for the agency to define whether overlapping deductions are allowed. If a deduction is configured to prevent overlapping deductions, a validation is displayed if a user tries to activate a deduction that exists on the case for an overlapping time period. This setting can be configured for all categories of deductions (applied, unapplied, and third party).

When a deduction type is associated with a benefit product, that deduction type is then automatically made available for selection at the case level. For more information about configuring deduction types, see the Cúram Deductions overview.

Configuring liability products

Two configuration settings apply to liability products only. A setting determines whether over allocation is allowed on the cases based on the liability product. Another setting determines whether reassessments on a liability product are either normal or reconciled. These settings trigger specific actions when liabilities are reassessed, depending on whether overpayments or underpayments are discovered during reassessment.

Enabling normal reassessment

A normal reassessment is a reassessment that does not attempt to reconcile any payments that are received against the liability amounts. Normal reassessment creates an overpayment or underpayment that is based solely on changing circumstances. Reconciled reassessment creates an overpayment or underpayment based on changing circumstances, and related payments that are received, and over allocation payments.

If the 'Normal Reassessment' setting is indicated and a case is reassessed, the system does not attempt to reconcile any payments that were received against the liability amounts. The basic calculation for normal reassessment is [Actual - Reassessed].

If the normal reassessment setting is not enabled and a case is reassessed, the case undergoes a reconciled reassessment. The system creates an overpayment or underpayment based on changing circumstances, related payments received, and over-allocation payments. The basic calculation for reconciled reassessment is [(Actual-Reassessed) - (Actual-Received)]. As part of reconciled reassessment, the assessment engine also allocates payments that are received and adjusts the unprocessed amounts on any over-allocation items. This processing does not occur as part of normal reassessment.

Enabling over allocation

An over allocation is the payment of an amount of money that is greater than the amount that was billed. If the Over Allocation setting is indicated for a liability product, amounts of money that are greater than the amounts that are billed to a participant can be allocated toward an outstanding liability. For example, if an employer is billed $100 and a payment is received from that employer for $120, the full $120 is allocated toward the $100, creating a $20 over allocation.

If the over allocation setting is not indicated, then the allocation cannot exceed the liability amount. Over allocations are used for products where the liabilities are usually only an estimate of what should be billed. For example, employer contribution liabilities are usually an estimate, as the employer tends to have more information than the agency about the number of its employees.

The Over Allocation setting is also used to determine how canceled payments are regenerated if the original payment included a deduction. If the setting is not enabled, when a payment is regenerated and the deduction that is made along with the original payment is more than the outstanding liability amount, an exact replica of the original payment will not be regenerated. The system deducts only the outstanding liability amount. If the setting is enabled, when a payment is regenerated and the deduction that is made along with the original payment is more than the outstanding liability amount, the system regenerates an exact replica of the original payment.

Configuring payment correction products

A payment correction is a preconfigured product that is used to bill participants who were overpaid or pay participants who were underpaid. Because it is a preconfigured product, it cannot be created in the administration application by using the product wizard. It is provided specifically for use in over and underpayment processing and can be maintained only through the administration application to meet the underpayment and overpayment processing needs of the agency.

An agency can configure benefit and liability products such that a payment correction case is created automatically each time that the system detects an overpayment or underpayment during case reassessment.

The configuration options that are available for the payment correction product are modeled on standard configuration options available for the benefit and liability product types. The payment correction case can serve as either an underpayment case or overpayment case:

- For settings that apply to multiple case types such as the case name, home page, and date settings, see “Configuring common case information” on page 1.
- For financial settings, see “Configuring case reassessment and adjustments financial processing” on page 11.

An administrator can also restrict access on payment correction cases by using the product security that is described in “Product access security settings” on page 14, configure approval checks, delivery patterns, milestones, and time constraints, and associate deductions to the payment correction product for use when the payment correction case represents an underpayment.

Configuring absence reasons

Absence reasons can be configured for benefit and liability products. You can define a list of the acceptable absence reasons that are relevant to a particular program.

For example, an agency can configure absence reasons that are acceptable for a client who is absent from attending a scheduled on-the-job training activity to prove work requirements for receiving Income Support. An example of an acceptable absence reason is 'sickness.'

By recording absence reasons, administrators can specify whether a client absence from a scheduled activity is payable or deductible. An absence reason can be configured to be both payable and deductible.

The Acceptable setting indicates whether the absence reason is acceptable. For example, an absence reason of 'family bereavement' might be acceptable for a client to be absent from a scheduled on-the-job training activity that they are required to attend for one program, but not for another.

Settings are also available to indicate whether an absence reason is payable or deductible. These settings are applicable only to IBM Cúram Provider Management processing. For more information about these settings, see the Cúram Provider Management.

Configuring appeal processes

A custom appeals process can be set up for the products that the agency provides. Appeal processes allow for the escalation of an appeal to a higher decision making body.

The three types of appeal are Hearing, Hearing Review, and Judicial Review.

Agencies can configure the order in which these types of appeal can be created when an appeal is created against a decision on a product delivery case. For more information about configuring appeal processes, see the Appeals business overview.

Configuring bonus payments

Bonus payments are once-off or non-standard payments that are issued to benefit recipients. For example, a back-to-school bonus that is given to a parent to help with the expense of new school materials.

The options available in the Bonus Type drop-down list are associated with the BonusTypeCode code table, which lists the type of bonus that is being paid. A new type can be dynamically added to this code table and published in system administration.

The Start Date and End Date settings are used to define the period during which the bonus is to be applied to cases based on this product.

The Bonus Amount setting is used to determine the bonus amount to be paid to benefit recipients. For example, a bonus payment of $50 might be paid to each participant over 65 to assist with fuel costs.

Configuring time constraints

Time constraints are the time limits applied to products. For example, a time constraint can be added to a product that defines the maximum amount of time after an eligibility decision is reached that a claimant has to appeal the decision.

The Number of Days setting is used to determine the maximum period in days that applies to the selected time constraint type.

For example, if the maximum number of days is 30 and the time constraint type is 'first appeal', the system sets a timeframe of 30 days within which a participant can make a first appeal on a case decision. If the time limit does not exceed the maximum period, the appeal is timely. If the time limit exceeds the maximum period, the appeal is untimely.

The Constraint Type setting works along with the Number of Days setting. This is the constraint type that relates to time limit functionality. When a type is selected, the system applies a time limit to the product that is governed by the number of days entered. For example, if the 'first appeal' type is selected, the 'first appeal' time limit is applied.

The options available in the Constraint Type drop-down list are associated with the ConstraintType code table. A new value can be added to this code table and published as part of system administration. The default value is 'first appeal'. The From Date setting is used to define the date from which the product time constraint is effective.

Configuring products to deliver services

A service can be configured to use product delivery processing to deliver the service. These types of services are a combination of the features of a standard product delivery and a standard service delivery.

Using product delivery processing allows caseworkers to use product delivery features such as evidence management, eligibility determination and financial processing for the service along with functionality that is specific to service deliveries, such as provider selection and provider enquiries.

For information about service and referrals, see “Configuring services and referrals” on page 20.

Services that use product delivery processing are configured in Cúram Provider Management (CPM) as service offerings. If an administrator configures a service offering to use product delivery processing, a product must also be specified. The product that is specified must first be configured so that it is suitable to be delivered as a service.

Most configurations that relate to a standard product are also applicable to a product that is used to deliver a service. For example, case evidence, rules that determine eligibility, decision details, rate tables, and financial processing can all be configured for a product that is used to deliver a service in the same manner that they are configured for a standard product delivery.

The following configuration options are specific to a product that is delivered as a service rather than as a standard product delivery. For more information about configuring service offerings and the different delivery types available for a service offering, see the Cúram Provider Management.

Configuring the display of product delivery information

When a caseworker creates a service for a client, a service delivery is created by the system and is displayed to the caseworker. If the service uses product delivery processing, a product delivery is also created in the background by the system, to handle the eligibility and financial processing for the service. Any product delivery functionality that is related to the eligibility and financial processing such as financials, determinations, and evidence is automatically displayed at the service delivery level and can be viewed by a caseworker in the context of that service delivery.

An administrator can enable or disable the display of the product delivery that is associated with the service in the case lists. To ensure that the product delivery is not returned in any search or case listing, disable the following product settings:

- Display in Participant Programs List
- Display in Curam Universal Access
- Display in My Cases Filter
- Display in Case Search Filter

Assigning a delivery pattern

A default delivery pattern is assigned to every product on creation. The delivery pattern defines how and when the product is delivered in the form of a payment.

Services can either use product delivery nominee functionality or invoice nominee functionality. If a service uses product delivery nominee functionality, payments in respect of the service are made to the product delivery nominee according to the nominee's delivery pattern. If the nominee does not have a specified delivery pattern, then the product's delivery pattern is used by default. If a service uses invoice nominee functionality, CPM determines the nominee and uses that nominee's default delivery pattern (rather than the product's delivery pattern) to issue payments in respect of the service.

Configuring open-ended cases

The Allow Open Ended Cases setting dictates if an expected end date must be defined for services that are based on this product.

For services that are configured to use product delivery processing, if this setting is not specified for the product, then you must specify an end date when you create the service delivery. When the end date is reached, financials that are issued in respect of the product delivery are automatically stopped.

Configuring deductions

Both new and existing deduction types can be associated with a product that is used to deliver a service. Deductions can then be created for any nominees that are defined for services that are based on that product.

Deduction functionality for products is only available to services that use product delivery nominees. This is because the nominee for the service might be the client. Any deductions in respect of services that use product delivery nominees can be managed by caseworkers at the service delivery level. If a service does not use product delivery nominees, that is, it uses invoice nominees, the nominee in most instances is a provider or provider group. Deductions in respect of these nominees are managed as part of CPM through provider financials and are not visible at the service delivery level.

Associating products with categories

An administrator can assign products to integrated case categories so that cases based on those products can be added to the integrated case. For products that are used to deliver services, you do not need to associate an integrated case type with the product. This is because the user adds a service, rather than a product, to the integrated case. The product delivery is created in the background to handle the eligibility and financial processing for the service, and is not used as a product in its own right.

Configuring payment groups

Agencies can use payment groups to group products or programs for which the payments need to be rolled up during financial processing. The system rolls up the payments of products that belong to a payment group into a single payment provided the conditions for roll-up are satisfied.

The payments for all remaining products that are not associated with any payment group are then rolled up into a single payment. For more information about the criteria for rolling up the payments, see Determining the roll-up of financial instruction line items.

Creating payment groups can be useful in a scenario where the agency does not want payments from various programs to be rolled up into a single payment instruction.

For example, James Smith is eligible to receive $100 per week from the Child Support program, $150 per week from the Income Support program, and another $100 from the Carer's Allowance program. If no payment groups are configured, these payment instruction line items are rolled up into one payment instruction of $350. However, if the agency does not want the Child Support payments to be rolled up with the payments for the Carer's Allowance Program, they can prevent this payment from occurring by creating a payment group.

In this scenario, the agency can create a payment group and add the Income Support and Carer's Allowance programs to it. The system then generates two payment instructions - one for $100 for the Child Support program and another for $250, which includes the rolled-up payments from the Income Support and Carer's Allowance programs.

Configuring services and referrals

Agencies can configure services and referrals to be created and delivered at the integrated case level.

Services for service delivery and referral processing are configured in IBM Cúram Provider Management as service offerings. Service offerings are the templates on which individual services that are created at the integrated case level are based.

For more information about configuring services and referrals, see “Configuring products to deliver services” on page 18 and the Cúram Provider Management.

Configuring services

Administrators can configure the information that can be recorded by a user when they create and manage services that are based on a particular service offering. For example, whether the service is appropriate for multiple clients or whether the automatic creation of multiple service deliveries is allowed.

Individual settings also exist on the service offering that dictate if a provider or provider type is mandatory or optional when a user creates a service.

Administrators also can define whether a service frequency and client participation frequency and duration can be specified on service creation. A default service frequency also can be specified and this frequency can be overridden manually by a user.

Additionally, a number of settings that dictate the delivery and the financial processing for the service are provided. An administrator can specify whether a user must record the required number of units of a service that must be delivered to a client. A setting also indicates whether a user can specify an authorized rate. If this parameter is set, a user can manually override the rate to be paid to a provider for providing the service. Administrators also can define the delivery type for the service that dictates what financial processing is used.

Configuring service provider selection

The following settings determine how a service provider is selected at the case level. Options determine whether a specific provider must deliver the service or any provider of a specific provider type must deliver the service.

Table 2. Options for configuring provider and provider type selection

Setting	Description
Provider Mandatory	The user must select a provider. The caseworker is not given the option to select a provider type.
Provider Type Mandatory	The user must select a provider type. The caseworker is not given the option to select a provider.
Provider or *Provider Type Mandatory*	The user is given the option to select either a provider or a provider type. Either a provider or a provider type must be selected.
Provider or *Provider Type Not Required*	The user is not presented with the option to select a provider or a provider type.
None Specified	The user is given the option to select a provider or a provider type. Provider or provider type is optional.

Configuring service inquiries

Two system administration properties dictate whether email notifications need to be sent in respect of service inquires.

- Use the **curam.serviceenquiry.SendProviderEmailNotifications** system administration property to indicate whether email notifications are sent to a provider when a service inquiry is created or updated that relates to one of its services.
- Use the **curam.serviceenquiry.sendCaseWorkerEmailNotifications** system administration property to indicate whether email notifications are sent to the user who owns the service inquiry when a provider responds to the service inquiry.

Configuring referral services

You can define the information that can be specified when a referral is created, such as whether the referral service is appropriate for multiple clients. You can also configure the notification templates that are used to generate the referral notifications that are issued to the client and the provider.

The application also provides a strategy that determines the notification mechanism that is used by the system to send notification letters automatically based on the participants' preferred communication methods and communication exceptions if they exist.

Configuring notification templates

The templates on which referral notification letters are based are configured in service administration and then associated with a referral service. If a template is not configured, a default pro forma template is provided and used for letter creation. Default templates are also configured in system administration.

Configuring follow-up notifications

- Use the **curam.referral.FollowUpApplicableForOrganization** property to indicate whether follow-up is required for referrals.
- Use the **curam.referral.SendEmailNotifications** property to indicate whether email notifications are sent to clients and providers.
- Use the **curam.referral.EscalationFollowUpWarningDays** property to indicate the number of days after which a supervisor notification needs to be sent.

Configuring a notification mechanism

The application provides a strategy that is used by the system to determine the means by which notification letters are sent to providers and clients. The strategy works with any communication exceptions and preferred communications that are recorded for the provider, the client or both during registration.

If hardcopy and email are communication exceptions, the client and provider cannot be notified automatically. If a provider or client has no communication exceptions, notification letters are sent based on the client's, the provider's or both preferred communication method.

If email is the preferred communication method and an email address is recorded for the client and provider, the letters are attached to emails and sent. If an email address is not recorded, or hardcopy is the preferred communication method, the system sends the letter to a printer for posting. The default printer that is specified for a user in administration is used as the target printer. For more information about user administration, see the Cúram Organization Administration Guide.

Note: If multiple clients are listed on the referral, each client receives a notification.

If no preferred communication is specified for the provider and client and both address and email details exist, agencies can use two system administration properties to specify whether the client and provider letters are to be sent either by email or sent to a printer for posting.

- Use the **curam.referral.DefaultClientCommunicationMethod** property to indicate whether provider letters are sent by email or sent to a printer for posting.
- Use the **curam.referral.DefaultProviderCommunicationMethod** property to indicate whether the client letters are sent by email or sent to a printer for posting.

Configuring case evidence and rules

Case evidence and rules are used to determine client eligibility and entitlement for the agency's products. Evidence is configured in administration as evidence types. Once configured, evidence types are associated with case types so that they can be captured at the case level. Evidence types are then linked to rules and rules are assigned to products to enable the system to determine client eligibility and entitlement.

To capture evidence types at the case level, each evidence type must first be configured, then activated, and associated with a case type. Associating an evidence type with a case type allows that evidence type to be captured at the case level for any cases that are based on that case type. Administrators can configure dynamic and non-dynamic evidence types, associate evidence types with case types, configure rule sets, design rules, and assign rules to products.

Note: EvidenceFlow is deprecated. For more information, see Deprecated features.

When an evidence type is configured to be captured on an integrated case type, that evidence can be captured on any integrated case of that type and is also used when determining eligibility and entitlement for product deliveries within the integrated case.

Administrators can also design and link rules that determine eligibility and entitlement to the evidence types during rules definition and assignment.

Configuring evidence types

Administrators create a new dynamic evidence type by configuring its name, logical name, effective period, and a group name that is used to contain security identifiers for the evidence type.

An effective from date allows flexibility for the evidence type to change over time. For example, an agency might configure an income evidence type to capture income-related information. Due to a change in legislation, the agency needs to update the income evidence to capture an additional piece of information. The agency creates a new version of the income evidence effective from the date of the legislation change. A version history of the evidence type records that are effective over time is automatically stored so the correct version of the evidence is available to the caseworker.

Basic information about each evidence type is defined during administration. This information includes the name of the evidence type, its logical name, and the name of the related group of security identifiers that are generated in respect of the evidence type.

Administrators can configure a group name to group security identifiers as needed. To grant a particular user read, create, and maintain rights for an evidence type, the administrator must add the group name to the user's role. For more information about assigning security identifiers to users, see User security profiles .

The Dynamic Evidence Editor

After basic information is defined about the evidence type, an initial version of the evidence type is created by default. Administrators can modify the evidence type metadata by using the Dynamic Evidence Editor to design the evidence pages that relate to the evidence type. In general, the editor is used to:

- Define all of the attributes that appear on the evidence pages.
- Define all of the evidence pages that relate to the evidence type.
- Set up validations on each evidence type. For example, for disability evidence if a disability type of ‘Acquired Brain Injury’ is entered, then the category of the brain injury must be entered.
- Set up parent and child evidence relationships. For example, Alien Sponsor evidence can be configured to be child evidence of Alien evidence.
- Design how the evidence page looks in the user interface. For example, an administrator can drag fields on to an application page cluster, or display one pane on a particular evidence page but not another.

For more information about configuring dynamic evidence by using the Dynamic Evidence Editor, see Configuring evidence.

Activating dynamic evidence

After an evidence type and its related pages are defined, the evidence type is activated to make it available to be associated with case types. Activating an evidence type also generates corresponding rule classes that are based on that evidence type, domain definitions, security identifiers, and evidence property files, and validates the evidence type data.

The rules classes that are generated allow rules developers to refer to the evidence attributes and enables them to write eligibility and entitlement legislation rules by using the CER (Cúram Express Rules) Editor.

Associating evidence types with cases

After an evidence type activated, that evidence type can be associated with a case. Dynamic evidence types can be associated with multiple integrated cases and products simultaneously.

As part of associating an evidence type with a case, the evidence type category and sort order is specified. A quick link setting is provided which allows an administrator to specify whether the evidence type and the category to which it is assigned is available as an item in the preferred menu list on the evidence workspace. The preferred menu list allows users quick access to the evidence types they frequently need to access daily.

The sort order is used to determine the order in which the evidence type is listed in the evidence workspace. If a sort order is not specified, the evidence type is listed alphabetically.

Evidence types are displayed in the evidence dashboard that is used by users to capture evidence. This display allows for access to the evidence pages associated with a particular case type. Each evidence category can contain one or more configured evidence types.

Each evidence type is associated with an integrated case type or product delivery case type. If the evidence type is to be managed at the integrated case level, it is associated with an integrated case type where it can be captured on the integrated case at the case level and also reused by any product deliveries within that integrated case. If the evidence type is to be managed from a stand-alone product delivery case, it is associated with the product that governs that case.

The following table describes the available configuration options for associating a dynamic evidence type with an integrated case type or product delivery case type:

Table 3. Evidence Association Configuration Settings

Configuration Setting	Description
Evidence Type	Indicates the preconfigured evidence type to be associated with the case type.
Category	Indicates the evidence category to which the evidence type belongs. For example, the 'Earned Income' evidence type belongs to the 'Income' evidence category. The evidence categories available for selection are associated with the EvidenceCategory code table. A new evidence category can be added to this code table and published as part of system administration. To provide flexibility, the same evidence type can belong to multiple categories. For example, the 'Earned Income' evidence type can belong to the 'Income' evidence category and the 'Household' evidence category. These evidence categories are displayed within the evidence workspace and are used to define the subset of evidence types that can be captured on a case. Each category is available as a filter option when a caseworker captures new evidence for that evidence type within the evidence workspace. For example, when capturing new evidence, a caseworker can filter the Category drop-down list by 'All'. The system automatically displays all evidence types that are associated with any category within administration.

Table 3. Evidence Association Configuration Settings (continued)	
Configuration Setting	**Description**
Sort Order	Indicates the order in which the evidence type is listed in the evidence workspace. For example, the sort order for the Pension Fund evidence type is set to 1 and the sort order for the Real Estate evidence type is set to 2. When a caseworker views the evidence types for a category to create new evidence within the evidence workspace, the Pension Fund evidence type is listed first and the Real Estate evidence type is listed second. By default, evidence types are listed in alphabetical order.
Quick Link	Indicates whether the evidence type is available as a 'Preferred' filter option in the evidence workspace. For example, if this setting is enabled for the 'Earned Income' evidence type, the 'Earned Income' evidence type is returned when a caseworker selects the 'Preferred' category filter option when they are capturing new evidence.

Enabling evidence sharing on evidence types

Evidence sharing allows evidence to be shared between cases.

Administrators can choose which evidence types to share and can enable or disable each evidence type for sharing on each case type. For example, an integrated case type can have several evidence types that are not all suitable for sharing, therefore only some evidence types are enabled.

For evidence sharing to work at the case level, agencies must also install IBM Cúram Evidence Broker. For more information, see Sharing evidence by using the evidence broker.

Configuring evidence approval checks

Evidence approval checks are used to determine whether the evidence that is created or modified by a user can be approved automatically or whether it requires manual approval by a supervisor. An administrator specifies the percentage of evidence changes that require manual approval and the remainder get approved automatically.

Evidence approval checks can be set up at the organization unit, user, position, and product levels. At the organization unit, position, and user levels, it must be specified whether the checks apply to a single product or all products. When the user submits evidence for approval, the system checks the user's evidence approval check setting, then checks the evidence approval settings for the user's organization unit. Then, the system checks the evidence approval setting at the product level. If at any point the system determines that the evidence requires approval, the evidence approval is assigned to a case supervisor for approval.

Defining rules and configuring rule sets

The Cúram Express Rules (CER) language can be used to convert legislation directly into rules that can determine client entitlements. CER also provides a representation of rules that allow users to understand how legislation impacts the decisions made on client entitlement.

Legislation is complex so the rules that express that legislation are complex. To make the designing of rules more efficient, tools can assist users, rules designers, and product administrators to work together to dynamically design rules, link these rules to a client's information, and assign them to products to determine client eligibility.

Before the system can determine client eligibility and entitlement, rules that are applied to client information must be designed as rules classes by using the CER Editor. Rule sets are configured to contain these rules, which must then be linked to the evidence types, non-evidence entities, and the products for which eligibility is determined. Configuration options support the rule set configuration, rules design, and the assignment of rules to evidence types and products.

Dynamic rules design - The CER Editor

The CER Editor is a dynamic tool that is used to define and validate CER rules. The main feature of the CER Editor is the canvas in which rules expressions are designed. This canvas provides a visual representation of rules that can be understood by rules designers and business users.

New rules expressions can be created in this canvas by using a menu bar that contains common rules logic, such as an 'if-then' statement that determines the age limit of a child, and custom logic for agencies.

For example, the CER Editor offers quick ways to create rules expressions that relate to household and financial units. In addition to the CER Editor canvas, short-cuts are provided to create new rules classes, group these classes in folders, and to validate rules expressions.

Configuring and categorizing rule sets

Administrators can configure rule sets by creating new ones or modifying existing ones from the list of rule sets that are currently active on the system. As part of rule set creation, the rule set name and display name are specified. The rule set name is the technical name of the rule set and cannot be modified. The display name of the rule set can be modified and is the name of the rule set that is displayed to an administrator within the administration application.

Each new rule set can be associated with an existing rule category by selecting it from existing categories. Administrators can create new rules categories. A name and description is added for each new rules category. To modify an existing rule set, administrators can search for the rule set by filtering it by category by using the Category drop-down list. The required rule set can then be modified from the list of rule sets returned. After a rule set is configured, the rules classes in the rule set and their corresponding attributes are defined or modified, typically by a rules designer, with the Cúram Express Rules Editor. The rule set is then validated and published.

For more information about creating and editing rules with the CER Editor, see the Working with Cúram Express Rules Guide and the Cúram Express Rules Reference.

Linking rules to evidence with a rules data configuration

For rules to be applied to a client's information, those rules must be linked to the evidence and non-evidence entities that are used to capture that information. For example, rules for determining income amounts would be linked to the income evidence type. Rules data configurations are used to link the data that is needed to ascertain eligibility and entitlement with the rules that are used to determine it.

A rules data configuration links a rule class to an evidence type or non-evidence entity. A rules data configuration is an XML file that can be maintained dynamically. A new rules data configuration can be added dynamically. Existing rules data configurations can also be copied and edited as required.

A name must be specified for each rules data configuration and data is then added in XML format. For example, a data configuration can be added that links income assistance data to the 'Income Assistance Product Eligibility and Entitlement' rule set.

Each new or in-edit rules data configuration must be published for it to become active on the system. When published, the changes are taken into account when case reassessment next occurs within any product delivery case based on a product that uses the updated rules data configuration. For example, when reassessment is manually initiated within the case or financials are generated. Information about changes that occur is captured by the system in a rules data configuration, which is then used by batch processes that can be initiated to reassess all cases affected by the change.

Assigning Cúram Express Rules to products

For eligibility decisions to be created in product delivery cases based on a product, rules for determining eligibility and entitlement must be assigned to products. You can assign Cúram Express Rules to benefit and liability products that are configured for use with Cúram Express Rules.

Assigning rules to products

Rule sets that are created by using the Cúram Express Rules (CER) editor can be categorized and assigned to products. Rule set categories provide administrators with a facility for filtering and sorting rule sets.

Three types of rule sets apply to the determination of client eligibility and entitlement for products:

- Eligibility and entitlement rules are normally related to or defined by legislation. These rules are needed for the system to determine whether clients are eligible and their entitlements.
- Decision details rules are used to provide an explanation of how eligibility and entitlement were determined. Agencies can design these rules to display explanatory information to users. For example, a list of countable assets or a hyperlink to supporting information. Custom pages can be designed to display the information that is determined by these rules. Users can use this information to match their understanding of the clients information to the system's determination of eligibility and entitlement.
- Key decision factor rules are used to provide visibility on key pieces of information that changed as part of eligibility and entitlement processing. By assigning a key decision factor rule set to a product, the significant factors that were involved in determining the client's eligibility, such as a change in total household income, can be displayed in the graphical representation of the determination that is created as a result of the eligibility and entitlement rule set associated to the product. Users can use these decision details rules to understand more about a client's eligibility and entitlement determination.

Note: The key decision factors feature is deprecated. For more information, see Deprecated features.

Assigning a product period to a product

Rules are assigned to products by using product periods. The product period defines the timeframe over which rule sets are applicable for the product, such as the 2021 calendar year. Therefore, each product must have a product period associated with it in order to determine client eligibility and entitlement. An administrator can create one or more product periods for a product.

Product periods can be open-ended, that is, only one product period need be defined for a product, and can be applied indefinitely. Administrators must assign an eligibility and entitlement rule to a product period and can optionally assign decision details rule or key decision rules.

Administrators can also specify the value of product attributes that are used in eligibility determination processing. For more information, see “Configuring eligibility determination and decisions” on page 6. Only published rules are used to determine eligibility and entitlement within a case.

Note: The key decision factors feature is deprecated. For more information, see Deprecated features.

The following table describes the available product period configuration settings.

Table 4. Product Period Configuration Settings

Configuration Setting	Description
Name	The name of the product period. This field is mandatory.
Start Date	This setting is used to determine the date from which the product period is effective. This field is mandatory. The start date defaults to the current date on which the product period is created.

Table 4. Product Period Configuration Settings (continued)	
Configuration Setting	**Description**
End Date	The date to which the product period is effective. Note that if no product periods assigned to a product have an effective date that covers the lifetime of the product delivery case, the system will only determine eligibility up to the end date specified by the latest product period. For example, a product delivery case has a case start date of January 1, 2011 and an expected end date of Dec 31, 2011, and its associated product has only one product period with an end date of November 1, 2011. The system will determine eligibility for the client up to November. From November 1, 2011, the following decision will be displayed at the case level: Eligibility Could Not Be Determined.
Product Structure Rule	This setting is used to assign the Product Structure rule set that defines what objectives and objective tags a client is potentially entitled to receive to the product. This field is mandatory. The administrator searches for the product structure rule by selecting the rule set to which it belongs from the Rule Set drop-down list. The product structure rule is selected from the list of rules displayed. Note that the rule sets that are available within the drop-down list must first be configured. For more information on configuring rule sets, see "Defining rules and configuring rule sets" on page 25.
Eligibility/Entitlement Rule	The setting is used to assign the Eligibility/ Entitlement rule set that is used to determine client eligibility and entitlement to the product. This setting is mandatory. An eligibility/entitlement rule set must be assigned to the product over a period of time that covers the lifetime of the product delivery case in order for client eligibility to be determined. For any periods of time for which there is no eligibility/entitlement rule set assigned, the following decision will be displayed at the case level: Eligibility Could Not Be Determined. The administrator must search for the eligibility/ entitlement rule by selecting the rule set to which it belongs from the Rule Set drop-down list. The appropriate eligibility/entitlement rule is selected from the list of rules displayed. Note that the rule sets that are available within the drop-down list must first be configured. For more information on configuring rule sets see "Defining rules and configuring rule sets" on page 25.

Table 4. Product Period Configuration Settings (continued)	
Configuration Setting	**Description**
Key Decision Factors Rule	This setting is used to assign the Key Decision Factors rule set that is used to extract information about the significant factors that were involved in determining a client's eligibility to the product. This setting is optional. These decision factors will be displayed in the graphical representation of the determination displayed at the case level. The administrator must search for the key decision factors rule by selecting the rule set to which it belongs from the Rule Set drop-down list. The appropriate key decision factors rule is selected from the list of rules displayed. Note that the rule sets that are available within the drop-down list must first be configured. For more information on configuring rule sets, see "Defining rules and configuring rule sets" on page 25.

Assigning decision details rules to a product

In addition to assigning product structure rules, eligibility/entitlement rules, and key decision factors rules to products, an administrator can optionally assign one or more decision details rules to a product.

Decision details rules are used to provide a user-friendly explanation of how eligibility and entitlement was determined, to assist the user in understanding the results of the execution of the eligibility and entitlement rule set. There are two steps to assigning decision details rules to products.

The first step is to configure the display categories for the product that specify how the information displayed to the user should be grouped and ordered when viewing the details of how a particular decision was determined. For example, the explanation of determination results may be grouped into a summary category, a household income category, and a medical expense category so that the user can view summary information first on one page followed by detailed information about how household income and medical expenses impacted a decision on subsequent pages. Each product can have multiple display categories associated with it.

The following table describes the configuration options that are available when adding a new display category to a product.

Table 5. Display Category Configuration Settings	
Configuration Setting	**Description**
Name	The name of the display category. This name determines the label to be used on the page that explains the eligibility results.
Display Order	This setting is used to specify the left to right order in which the page displaying the information for a category will appear. For example, if the Summary display category is set to 1, the page which displays Summary information for a decision will be displayed first.
Display Page	This setting is used to specify the name of the `.jsp` page that is displayed for the category.
Category Reference	The user entered reference number that represents this display category.

The second step is to associate appropriate decision details rule to the product period. For each decision details rule, this is done by selecting the display category page on which decision details will be displayed and then assigning the rules used to determine the information that is displayed for the category. An administrator must first select a display category from the drop-down list of categories associated with the product. The administrator must then search for the decision details rule by selecting the rule set to which it belongs from the Rule Set drop-down list. The appropriate decision details rule is selected from the list of rules displayed. Note that the rule sets that are available within the drop-down list must first be configured. For more information on configuring rule sets, see "Defining rules and configuring rule sets" on page 25.

Publishing rules for products

When rules are initially assigned to products, they are in an in-edit state and must be published for the system to use them to determine eligibility and entitlement within a case. After rules are published, an administrator can edit existing product rules by making a copy of the published rules for a product, and editing the rules as needed.

Rule changes can then be published or discarded. If the changes are published, a snapshot of the previous version of the rules is taken and a new version is saved.

When published, the changes are used when case reassessment next occurs within any product delivery case based on the product. For example, when reassessment is manually initiated within the case or financials are generated. The system also detects a change to the product configuration, which is then used by batch processes that can be initiated to reassess all cases affected by the change. For an overview of how system-wide changes are applied to cases, see "Configuring eligibility determination and decisions" on page 6.

The administrator can also view a history of all versions of the rules that were associated to the product over time.

Assigning Cúram Rules to products

Rule sets that are implemented in Cúram Rules can be assigned to benefit or liability products that are configured for use with Cúram Rules. The process begins with selecting the rule set. The rule set is searched for and selected from the list of existing rule sets.

At least one rule set must be assigned to a product as a prerequisite to eligibility determination. Information that is recorded for the rule set assignment includes the name of the eligibility rule set to be assigned to the product and the period during which the rule set is assigned to the product. This period cannot overlap with the periods of existing product rule set assignments. One or more rule sets can be assigned to a product. However, only one rule set can be active during a period.

Rule sets that are implemented in Cúram Rules are maintained by using the Cúram Rules Editor. For more information about maintaining these rule sets, see the `Cúram Rules Editor Guide`.

Configuring ongoing case management

Administrators can configure ongoing case management by setting application properties. They can configure the case overview, the case search facility, case approval, case reassessment, case closure, and the number of entries to be displayed in the case transaction log. Case milestones can also be configured as part of ongoing case management.

Case search configuration allows the agency to control whether standard or enhanced search is used when the agency is searching for cases at the case level.

Case approval can be configured by using case approval checks. Administrators can also configure the automatic approval of cases by user role. Approval checks determine the percentage of cases that are reviewed by a case supervisor. For example, a case approval check can be set up such that 50 percent of cases that are submitted for approval require manual approval. The other 50 percent are approved automatically. At the organization unit and user levels, it must be specified whether the case approval

check needs to apply to a single product or to all products. Setting it at the product level governs all product delivery cases based on that product (across all users).

Configuring the case overview

You can show an **Overview** tab on selected integrated cases, which displays an Entitlements card by default. You must configure the **Entitlements** card to display information for the product delivery cases that you want to include on the card. You can customize the existing text label values for the **Overview** and the **Entitlements** card. You can also customize the data sources for existing data on the **Entitlements** card, or customize the destination application pages for Entitlements card links.

Adding new data fields to the **Entitlements** card or changes to text other than changing the existing text label values are not currently supported.

Customizing default text labels

If you want to change any of the default text label values on the **Overview** tab or **Entitlements** card, follow the standard process for changing text in the application by modifying properties files, see Externalized strings.

Configuring the case overview tab and Entitlements card

You can configure specified case types to show a card with entitlement and payments information from specified product delivery case types. An **Overview** tab with the **Entitlements** card is shown as the second tab on the configured case types. You must configure the case navigation files to include the new tab, and create overview tab XML configuration files to upload the configuration for each case type.

- "Updating the tab navigation to include the Overview tab" on page 31
- "Enabling GraphQL" on page 32
- Creating the overview tab configuration files
- Configuring the overview tab with build targets
- Adding the final configuration to the build

You can manually upload the case overview tab configuration files during development and test, and then set up the build time configuration.

Updating the tab navigation to include the Overview tab

For each product category where you want to show the case overview, update the overview tab configuration.

Procedure

1. Add a `navigation-page` element for the overview tab to each product category navigation file (`.nav`), ensuring that you set the `title` and `dynamic` attributes. For example, for Income Support.

   For example,

   ```
   <nc:navigation-page id="CaseOverview" page-id="Case_overviewPage"
   title="Page.Title.CaseOverview" dynamic="true" />
   ```

   For more information about `navigation-page` element and dynamic navigation, see Tab navigation navigation-page element and Tab navigation dynamic support.
2. Add a `DynamicNavStateLoader` to the `.nav` file, or use an existing `DynamicNavStateLoader`:

   ```
   <nc:loader-registry>
     <nc:loader class="curam.sample.SampleLoader"/>
   </nc:loader-registry>
   ```
3. Call the following code in the `DynamicNavStateLoader`.

```
// Only display the overview tab if enabled for this product category, for example Income
Support
CaseOverviewTabHelper.configureCaseOverviewTab(navigation, pageParameters);
```

Enabling GraphQL

Data that is displayed on the Entitlements card is populated by GraphQL queries. GraphQL is disabled by default, so you must enable GraphQL with a system property to enable data to be shown on the Entitlements card.

Procedure

1. Log into the application as a system administrator.
2. Select the **System Configurations** tab, expand the **Application Data** menu, and select **Property Administration**.
3. Set the **`curam.graphql.endpoint.enabled`** property to `true`.

Creating overview tab configuration files

For each integrated case type where you want to show the case overview, you must create an overview tab XML configuration file and use it to upload the configuration to the database. The configuration file defines whether the overview tab is shown and what cards are shown on the Overview. For example, for the Entitlements card, you must define which product delivery cases to include on the card, and the target destinations for any links on the card. For example, you might want to configure the Income Support integrated case to show the Cash Assistance and Food Assistance products.

Procedure

In an XML editor, create an XML configuration file as shown in this example.

```
<?xml version="1.0" encoding="UTF-8"?>
<overview-tab-config>
 <case-type-code>CT5</case-type-code>
 <product-category-code>PC4000</product-category-code>
 <show-overview-tab>true</show-overview-tab>
 <cards>
  <card>
   <card-name>Entitlements</card-name>
    <product-delivery-cases>
     <product-delivery-case>
      <product-delivery-case-name>PN4100</product-delivery-case-name>
      <links>
       <navigation-link>
        <name>home</name>
        <uim-page>CREOLECA_productDeliveryHome</uim-page>
        <parameters>
         <name>caseID</name>
        </parameters>
       </navigation-link>
       <navigation-link>
        <name>determinationHistory</name>
        <uim-page>ProductDelivery_listDeterminations</uim-page>
        <preferred-tabs>
         <tab>CREOLECAProductHome</tab>
        </preferred-tabs>
        <parameters>
         <name>caseID</name>
        </parameters>
       </navigation-link>
       <navigation-link>
        <name>payments</name>
        <uim-page>ISPProduct_listFinInstruction</uim-page>
        <preferred-tabs>
         <tab>CREOLECAProductHome</tab>
        </preferred-tabs>
        <parameters>
         <name>caseID</name>
        </parameters>
        </navigation-link>
```

```
        <navigation-link>
         <name>overPayments</name>
         <uim-page>ISPProduct_listBenefitOverUnderPayment</uim-page>
         <preferred-tabs>
          <tab>CREOLECAProductHome</tab>
         </preferred-tabs>
         <parameters>
           <name>caseID</name>
         </parameters>
        </navigation-link>
       </links>
      </product-delivery-case>
     </product-delivery-cases>
  </card>
 </cards>
</overview-tab-config>
```

Where

- `case-type-code` is the case type, such as CT5 for integrated cases, which is the code from the `CaseTypeCode` code table.
- `product-category-code` is the code from the `ProductCategory` code table. All products can be placed into one or more categories, such as PC4000 for Income Support.
- `show-overview-tab` is set to `true` to display the overview tab for the specified integrated case type and `false` otherwise.
- `product-delivery-case-name` is a product delivery case type that you want to include in the **Entitlements** card. For example, you might want to configure the Income Support integrated case to show entitlements and payments for the Cash Assistance and Food Assistance products. This value is the `NAME` of the product delivery case type from the `ProductName` code table. For example, PN4100 for Cash Assistance.
- `navigation-link` specifies the page that opens when you click a link on a card, such as **See determination history** on the Entitlements card. For each navigation link, you must specify a target page and, where needed, parameters or a preferred tab. To disable a link, remove the <`navigation-link`> element.
  - `name` is the name of the target page, such as the **Determinations History** page on the product delivery case.

    For example, for the Entitlements card you can configure four links:

    - `home` The home page for the product delivery case.
    - `determinationHistory` The determinations history for the case.
    - `payments` The payments details page. This link is relevant only to product delivery cases with payments.
    - `overPayments` The over and underpayments details page. This link is relevant only to product delivery cases with payments.
  - `uim-page` The name of the UIM page to include in the link.
  - `parameters` Any parameters that must be appended to the link. For example, the `caseID`.
  - `preferred-tabs` Where pages display on multiple tabs, you must specify the tab where you want the page to open, such as `CREOLECAProductHome` for the product delivery case home page.

Configuring the overview tab with build targets

During development and test, you can use the overview tab build targets to manually configure the case overview by uploading the configuration from the XML configuration files. These build targets provide a certain level of validation for the configuration files during development. You can download an existing configuration to an XML configuration file to modify and upload again.

Procedure

1. To upload a configuration from an overview tab XML configuration file, complete the following steps:

a) Go to `C:\IBM\Curam\Development\` and run the following command:

```
SetEnvironment.bat
```

b) Go to `C:\IBM\Curam\Development\EJBServer` and run the following build target to upload a configuration from an XML configuration file to the database.

```
build.bat case.overview.configuration.upload
-Dlocation=C:\test\CaseOverviewConfig_PC4000v1.xml
```

Where **`-Dlocation`** is the absolute path to a configuration file.

2. To download an existing configuration to an overview tab XML configuration file, complete the following steps:

a) Go to `C:\IBM\Curam\Development\` and run the following command:

```
SetEnvironment.bat
```

b) Go to `C:\IBM\Curam\Development\EJBServer` and run the following build target to download a configuration from the database to a file:

```
build.bat case.overview.configuration.download
-Dlocation=C:\test\CaseOverviewConfig02.xml -DcaseTypeCode=CT5
-DproductCategoryCode=PC4000
```

This example downloads the configuration for the integrated case with the case type CT5 and the product category code PC4000 from the database to a file called `CaseOverviewConfig_PC4000v2.xml`.

```
build.bat case.overview.configuration.download -Dlocation=CaseOverviewConfig02.xml
```

Adding the final configuration to the build

When the configuration files are complete and tested, you can set up the overview tab configuration to run at build time.

Procedure

1. Create the following folder structure in the `EJBServer/components/custom/data/` folder.

```
├── data
|   └── demo
|   └── initial
|   |   └── clob
|   |   |   └── SampleCaseOverviewConfig.xml
|   |   |   OVERVIEWTABCONFIG.dmx
```

2. Put your overview tab configuration files in the `clob` folder as shown.
For example, `SampleCaseOverviewConfig.xml`.
3. Create an `OVERVIEWTABCONFIG.dmx` file in the `initial` folder based on the following example:

Where

- The value of the CASETYPECODE attribute matches the value of `<case-type-code>` in `SampleCaseOverviewConfig.xml`.
- The value of the PRODUCTCATEGORYCODE attribute matches the value of `<product-category-code>` in `SampleCaseOverviewConfig.xml`.
- The value of the CONFIGXML attribute is the location of the overview tab XML configuration file in the file system.

```
<table name="OVERVIEWTABCONFIG">
  <column name="OVERVIEWTABCONFIGID" type="id" />
  <column name="CASETYPECODE" type="text" />
  <column name="PRODUCTCATEGORYCODE" type="text" />
```

```
  <column name="CONFIGXML" type="clob" />
  <column name="VERSIONNO" type="number" />
  <column name="LASTWRITTEN" type="timestamp" />
  <row>
    <attribute name="OVERVIEWTABCONFIGID">
      <value>2501</value>
    </attribute>
    <attribute name="CASETYPECODE">
      <value>CT5</value>
    </attribute>
    <attribute name="PRODUCTCATEGORYCODE">
      <value>PC4000</value>
    </attribute>
    <attribute name="CONFIGXML">
      <value>./data/initial/clob/SampleCaseOverviewConfig.xml</value>
    </attribute>
    <attribute name="VERSIONNO">
      <value>1</value>
    </attribute>
    <attribute name="LASTWRITTEN">
      <value>SYSTIME</value>
    </attribute>
  </row>
</table>
```

Customizing data sources for the Entitlements card

Data that is displayed on the Entitlements card is populated by GraphQL queries. While you cannot customize the GraphQL schema, because the Entitlements card expects certain attributes to be returned by the query, you can customize the data sources for individual data items.

- "Cards on the Overview tab" on page 35
- "GraphQL queries" on page 35
- "The GraphQL schema" on page 36
- "Viewing the GraphQL schema" on page 36
- "Data fetchers" on page 37
- "Customizing data sources for GraphQL queries" on page 38

Cards on the Overview tab

Each card is a separate JavaScript component that is based on the IBM Carbon Design System. Cards are bundled as `SPMComponents` and linked to the **Overview** tab as a field in the associated UIM file. Data for the card is supplied from a GraphQL endpoint that is deployed as part of the REST EAR. Each card builds GraphQL queries for the data that it requires based on the entities in the GraphQL schema. The GraphQL schema is linked to an underlying SPM facade by a Java™ class called a data fetcher.

GraphQL queries

GraphQL is a query language for APIs. Web clients build queries for the data that they need based on the underlying schema. The schema consists of a set of entitles that are linked based on the business relationships between them so that they form a graph. Each query represents an entry point to the graph. After a query is named and the correct parameters that are passed in, the client can specify which elements from the schema to include in the response.

The following example request names the `readBenefit` query and passes in the `benefit_id` parameter. The request specifies a subset of fields from the benefit type to include in the response:

```
{
  readBenefit(benefit_id:"-6574692506007502848"){
    id
    reference
    status {
      code
      description
    }
    product_name {
      code
```

```
      description
    }
    product_type {
      code
      description
    }
    period {
      start_date
      end_date
    }
  }
}
```

The GraphQL schema

The GraphQL schema is defined in a schema file that is called `schema.graphqls` in the following location:

```
%CURAM_DIR%/EJBServer/components/core/rest/grpahql/config
```

Where *%CURAM_DIR%* is the SPM installation directory, by default `C:\IBM\Curam\Development`.

The following example represents one of the types in the schema. All types are prefixed by `DOM` to show that they represent objects from the Domain model.

```
type Query {
    readIntegratedCase(case_id: GQL_ID!): DOMIntegratedCase
}


type DOMIntegratedCase {
        id: GQL_ID
        reference: String
        effective_date: GQL_Date
        registration_date: GQL_Date
        status: CodeItem @code
        type: CodeItem @code
        benefits: [DOMBenefit]
}


type DOMBenefit {
        id: GQL_ID
        reference: String
        type: CodeItem @code
        product_name: CodeItem @code
        product_type: CodeItem @code
        status: CodeItem @code
        effective_date: GQL_Date
        period: DOMPeriod
}
```

This sample shows that the `DOMIntegratedCase` type contains a set of fields and is also linked to an array of `DOMBenefit` objects. In this way, elements in the schema are connected in a graph and the requester can explicitly specify the data that they are interested in fetching.

Some fields types include:

- `GQL_ID` represents a unique identifier.
- `CodeItem @ code` represents a code table type.
- `GQL_Date` represents a CURAM date type.

Viewing the GraphQL schema

GraphQL can process certain queries, called introspection queries, that return details about the schema. Introspection queries are disabled by default and must be enabled. You can then use an IDE or browser

page to view the schema, which sends an introspection query and uses the response to display all details about the available queries.

About this task

In a development environment, the GraphiQL in-browser HTML page is automatically included in the REST application that runs on Tomcat. The GraphiQL HTML page is not included in the deployed REST ear, so the schema is not exposed in production environments.

Procedure

1. Enable GraphQL intersection queries.
   a) Log in to the application as a system administrator.
   b) Select the **System Configurations** tab, expand the **Application Data** menu, and select **Property Administration**.
   c) Set the **curam.graphql.introspection.enabled** property to `true`.
2. Build the REST application:
   a) Set the `CATALINA_HOME` environment variable to the home directory of the Tomcat installation. This variable is used to automatically deploy the REST application into Tomcat.
   b) From the `%CURAM_DIR%/EJBServer` directory, run the following command-line command:

      ```
      build rest
      ```

      `%CURAM_DIR%` is the Cúram installation directory, which by default is `C:\IBM\Curam\Development`.
3. You can view the GraphQL schema details with a browser at: `http://`**<server>**`:`**<port>**`/Rest/graphql.html`
   For example, `http://localhost:9080/Rest/graphql.html`

Data fetchers

Each type in the schema is linked to a facade by a data fetcher Java class. The data fetcher class is a wrapper that does any simple operations that are needed on the input parameters and then calls the facade.

```
public class BenefitListDataFetcher implements DataFetcher<List<DOMBenefit>> {


  @Override
  public List<DOMBenefit> get(final DataFetchingEnvironment env) throws Exception {


    /* assign values to the input struct from the GraphQL request parameters, which are
    * available from the DataFetchingEnvironment object.
    */
    final DOMCaseID domCaseID = new DOMCaseID();
    final DOMIntegratedCase domIntegratedCase = (DOMIntegratedCase) env.getSource();
    domCaseID.case_id = domIntegratedCase.id;


    final String facadeClassName = "DOMBenefitGQL";
    final String facadeMethodName = "listBenefitsByIntCase";


    final DOMBenefitList benefitList = (DOMBenefitList) GraphQLUtils
      .callServer(facadeClassName, facadeMethodName, inputStruct);


    return benefitList;
  }
}
```

In this example, the data fetcher class gets the parent object from the DataFetchingEnvironment to get the integrated case ID. If the case ID is passed into the request by the caller, then you can get the ID as follows:

```
domCaseID.case_id = env.getArgument("case_id");
```

A configuration file that is called runtime_wiring.yaml links the types from the schema to the DataFetcher classes. The wiring file contains wiring to both queries and to other fields in the graph.

```
Query:
  - name: readIntegratedCase
    data_fetcher: curam.core.graphql.datafetcher.IntegratedCaseDataFetcher

FieldLevelWiring:
  - object_type: DOMIntegratedCase
    field:  benefits
    data_fetcher: curam.core.graphql.datafetcher.BenefitListDataFetcher
```

The following fields are in the wiring file:

- name is the name of the top-level query.
- data_fetcher is the data fetcher class.
- object_type is the parent object.
- field is the field in the parent object.

In the previous example, the readIntegratedCase query is linked to the IntegratedCaseDataFetcher. As a list of benefits is returned as part of the DOMIntegratedCase object, the benefits field is wired to its own data fetcher called BenefitListDataFetcher.

Generally, fields that are linked to nested lists have their own data fetchers, but you can also wire an individual field to a data fetcher.

Customizing data sources for GraphQL queries

The Entitlements card relies on the underlying GraphQL schema, which is deployed in the REST EAR, to get the data to display on the card. As you cannot change the Entitlements card, you must not change the GraphQL schema as any changes can impact directly on the card and create issues. However, you can maintain the interface that is represented in the GraphQL schema, and change the source of the underlying data. This approach enables your customizations to be integrated into the schema so that they show up correctly in the card.

Before you begin

You must have access to an SPM Development Environment.

Procedure

1. Identify which schema elements to customize.

   First, identify the elements in the schema for which you intend to customize the data source. Then, look in the runtime wiring to find the name of the data fetcher that currently links the schema element to the existing backend facade.
2. Create a custom SPM facade.

   To customize the source of the data, you must create a custom SPM facade in your own component. The new facade must match the same interface that the schema defines based on the existing facade. The input struct must contain attributes to match the request parameters, and the output struct must match the name and type of the query response object that is defined in the schema.

   The new custom facade can retrieve data from any part of the system if it complies with the same interface. Any AppExceptions thrown by the facade are displayed directly on the card.
3. Create a custom data fetcher Java class.

The next step is to create a data fetcher to wrap the new SPM facade so that it can be linked to from the schema. The job of the data fetcher is to parse the input from the GraphQL request and map it to `structs` so that the facade can be called.

Use the provided example to create the data fetcher, and save it into your custom component at the following location:

```
%CURAM_DIR%/EJBServer/components/<COMPONENT_NAME>/source/**/graphql/datafetcher
```

Where *%CURAM_DIR%* is the SPM installation directory, by default `C:\IBM\Curam\Development`.

4. Add an entry to the GraphQL runtime wiring configuration yaml file.

   The schema element is linked to the existing facade by the runtime wiring file. You must introduce your own version of this wiring file in your custom component. This wiring overrides the matching wiring in the existing file on a line-by-line basis.

   Create a file called `runtime_wiring.yaml` in your component at the following location:

   ```
   %CURAM_DIR%EJBServer/components/<COMPONENT_NAME>/rest/graphql/config
   ```

   Where *%CURAM_DIR%* is the SPM installation directory, by default `C:\IBM\Curam\Development`.

   At build time, `runtime_wiring.yaml` files from all server components are merged into a single `runtime_wiring.yaml` file. Where duplicate entries are found for queries or fields, the entry in the component with the higher component order takes precedence. The component order ensures that entries in a custom component overwrite corresponding entries in the default components.

5. Build the REST application, which includes the GraphQL artifacts. You must build the full REST application at least once. However, if you change only the GraphQL Java data fetcher classes or the `runtime_wiring.yaml` file, then you can rebuild just the GraphQL artifacts:

   a) Set the `CATALINA_HOME` environment variable to the home directory of the Tomcat installation. This variable is used to automatically deploy the REST application into Tomcat.

   b) To build the full REST application, enter the following command from the `%CURAM_DIR%/EJBServer` directory:

   ```
   build rest
   ```

   `%CURAM_DIR%` is the Cúram installation directory, which by default is `C:\IBM\Curam\Development`.

   c) To rebuild the GraphQL artifacts only, enter the following command from the `%CURAM_DIR%/EJBServer` directory:

   ```
   build graphql
   ```

Configuring the enhanced case search

By default, the standard case search is enabled. An enhanced case search that depends on the Generic Search Server can be enabled within system administration using application properties. Administrators can enable or disable the enhanced case search as required.

Note: The Generic Search Server that is required by the enhanced case search is deprecated.

Two application properties must be enabled for the enhanced case search to work:

- The `curam.lucene.luceneEnhancedSearchEnabled` application property enables the enhanced search functionality in the application. By default, this property is set to NO. An administrator can enable enhanced searching by setting this property to YES.
- The `curam.lucene.luceneEnhancedCaseSearchEnabled` application property enables the enhanced case search. By default, this property is set to NO. An administrator can enable the enhanced case search by setting this property to YES. When this property is enabled, a client first and last name can be specified as search criteria on the case search page.

An additional application property is available that makes changes to searchable data available to the enhanced search:

- The `curam.lucene.luceneOnlineSynchronizationEnabled` application property enables an event publishing mechanism that recognizes changes to searchable data. By default, this property is set to NO. This property must be set to YES in order for changes to searchable case information to be recognized and picked up by the search server.

Configuring case approval

Administrators can configure case approval for product delivery cases only.

Configuring case approval checks

Agencies can configure case approval checks on all cases that are governed by a product. Setting case approval checks at the product level governs all cases that deliver that product. The Percentage setting is used to specify the percentage of cases based on that product that need to be reviewed by a case supervisor. For example, if 40% of cases require manual approval, the other 60% are automatically approved.

The Estimated Cost setting is used to specify the estimated cost of the cases submitted for approval. All cases that cost more than the estimated cost that is specified on the configured case approval check are sent to a case supervisor for approval. This setting works with the Percentage setting. Any case at or under this estimated cost is checked for approval only if it falls within the percentage of cases to be checked.

For example, if a percentage of 20 and an estimated cost of $200 are specified, only 20% of submitted cases that cost $200 or less need to be manually checked for approval. All of a product's cases over the $200 estimated cost must be checked for approval.

Case approval checks can also be set up at the organization unit and user level. At the organization unit and user levels, an administrator can specify whether the case approval check applies to a single product or to all products. If all products are indicated, then all of the user's cases that are submitted for approval need to be checked by a case supervisor. If the case approval check applies to a single product, only cases that are governed by that product require approval. The case approval check settings for a product are the last step in the system's evaluation of whether a case requires approval. When a case is submitted for approval by a user, the system first checks the user's case approval check setting, then checks the case approval settings for the organization unit that the user belongs to. Then, the system checks the case approval setting at the product level. If at any point the system determines that the case requires approval, the case is assigned to a case supervisor for approval.

Configuring automatic case approval by user

Use the `curam.case.productdeliveryapproval.automaticSupervisorCaseApproval` application property to indicate whether the system automatically approves product delivery cases that are submitted for approval by case supervisors. By default, this property is set to YES.

If you set this property to NO, the system does not automatically approve product delivery cases that are submitted for approval by the case supervisor. On submission, the case supervisor must manually approve each case to progress it to the next stage in its lifecycle.

Configuring the delayed processing display

Delayed Processing Pending is a case status that indicates to the user that substantial system processing is happening when an online process runs. Delayed processing means that a user does not have to wait for a process to complete before they continue with other processing.

The **`curam.custom.delayedprocessing.dpdisplayinstatushistory`** application property is used to determine whether the **Delayed Processing Pending** status is displayed in the Case Status History for a product delivery case. The default value of this property is YES meaning that the status is displayed each time that delayed processing occurs.

The **Delayed Processing Pending** status is temporarily displayed on the Product Delivery Home page regardless of the value set for this property. This status is displayed until the delayed processing is complete. The user then knows that other processing can occur while the product delivery is in the delayed processing pending state.

Configuring case suspension and case closure

Use system application properties to configure case suspension processing and case closure processing.

One system application property controls case suspension processing:

- Use the `curam.miscapp.payuptosuspendeddate` application property to determine whether payments are issued up to the date of suspension after a case is suspended. The default value of this property is YES meaning that payments are issued up to the date of suspension. If this property is set to NO, no payments are issued after the case is suspended.

Three system application properties control case closure processing:

- Use the `curam.case.closeCaseWithTasks` application property to determine whether cases that have outstanding tasks that are associated with them can be closed. This property is applicable to product delivery and integrated case closure. The default value of this property is YES meaning cases that have outstanding tasks that are associated with them can be closed. If this property is set to NO, cases cannot be closed until all outstanding tasks for those cases are closed.
- Use the `curam.case.productdeliveryactivationeligibility.ineligibleautoclosecase` application property to determine whether ineligible cases are automatically closed. The default value of this property is NO meaning that an ineligible product delivery case is activated in the same way as an eligible product delivery. Reassessment occurs if the evidence changes on the ineligible case, which might result in an eligible decision.
- Use the `curam.miscapp.payuptocloseddate` application property to determine whether payments are issued up to the date of closure when a case is closed. The default value of this property is YES meaning that payments are issued up to the date of closure. If this property is set to NO, financial components are closed and payments are no longer issued.

Configuring case milestones

Milestones are used to track the completion of significant events or tasks during the lifecycle of a case. For example, to complete an initial medical exam within 7 days of the creation of a case. All case milestones are based on an associated milestone configuration. To support the manual and automatic creation of milestones within a case, you must associate a milestone configuration with the case type (product delivery, integrated case, or screening type).

Each milestone has an expected start and end date for when an event is scheduled, and an actual start and end date to track the actual milestone dates. For milestones that are created automatically, the expected start date and expected end date cannot be changed unless the milestone is configured to allow the dates to be changed. A milestone can be configured to require that a milestone waiver request is submitted for approval before the expected start date and expected end date can be changed.

A milestone waiver request can then be set up to allow the milestone expected start and end dates to be changed for an automatically created milestone.

Milestone waiver request approval check settings determine the percentage of submitted waiver requests for a milestone of a particular type that need to be reviewed by a case supervisor. For example, an approval check can be set up on a milestone that requires 60 percent of all submitted requests to be approved, and 40 percent to not require approval.

Milestone waiver request approval checks can be set up at the organization and user level for all milestone types or for a milestone of a particular type. User configuration settings take precedence over organization unit settings. When a waiver request is submitted for approval by a user, the system first checks the user's milestone waiver request approval check settings. Then, the system checks the milestone waiver request approval check settings for the organization unit to which the user belongs. The system can determine at any point in this process that the milestone waiver request requires approval.

New milestones are configured and then associated to case types, within which the milestones are then created. When associating a milestone to a case type, development effort might also be needed, see Developing with milestones.

Configuring recent case actions

You can set the number of days that cases are displayed on a user's lists of recently assigned, viewed, and approved cases.

The following properties are available:

- Use the **curam.user.caserecentassigned** application property to determine the length of time (in days) within which the system determines cases to be recently assigned. For example, if this value is set to 7, any cases that were assigned to a user within the last 7 days are displayed on that user's list of recently assigned cases. The default value of this property is 7.
- Use the **curam.user.caserecentviewed** application property is used to determine the length of time (in days) within which the system determines cases to be recently viewed. For example, if this value is set to 7, any cases that were viewed by a user within the last 7 days are displayed on that user's list of recently viewed cases. The default value of this property is 7.
- Use the **curam.user.caserecentapproved** application property to determine the length of time (in days) within which the system determines cases to be recently approved. For example, if this value is set to 7, any cases that were approved for a user within the last 7 days are displayed on that user's list of recently approved cases. The default value of this property is 7.

Configuring the case transaction log

Administrators can configure the maximum number of entries that are displayed within the case transaction log on the case home page. The value that is set for this property applies to both integrated case and product delivery case transaction logs.

Use the `curam.casetransactionlog.nooftransactions` application property in system administration to determine the maximum number of entries that are displayed within the case transaction log on the case home page. The default value is 10. If the number of case transactions exceeds the maximum number set, only the most recent transactions are displayed in the transaction log.

Configuring how to display case clients on case list pages

Use the `Display Case Clients on Case List Pages` application property to hide the Primary Client column and display a Clients column on pages.

The property applies to the following pages:

- **My Recently Assigned Cases**
- **My Recently Approved Cases**
- **My Recently Viewed Cases**
- **My Cases**
- New Case Query results
- Case Search results
- Saved Case Query results

When the Clients column is configured to be displayed, the following case members are listed:

- All active case members with a case participant role of 'Member' or 'Primary Client', including those with an end date, for integrated cases.
- All active case members with a case participant role of 'Member' or 'Primary Client', including those with an end date, for product delivery cases.

For all other case types that might be displayed, the default behavior is also to display all active case members with a case participant role of type ‘Member’ or ‘Primary Client’, including those with an end date.

Note: For case types that do not assign a case participant role of type ‘Member’ to any case participants, only the case participant with a case participant role of type 'Primary Client' is displayed.

Case member names are sorted in alphabetical order and the case type determines the case members that are displayed. Case member names do not include a hyperlink.

The default value of the `Display Case Clients on Case List Pages` property is **False**. If you set the value to **True**, multiple case members are displayed.

Enabling hook points

If the `primaryclient.display.caselistpages.caseclients` application property is set to **True**, you can use the `curam.core.hook.impl.CaseClientsPopulationHook` hook point to customize the list of clients to display in the following case list pages:

- **My Recently Assigned Cases**
- **My Recently Approved Cases**
- **My Recently Viewed Cases**
- **My Cases**
- New Case Query results
- Case Searches results
- Saved Cases results

You can enable the hook point through the standard Google Guice dependency injection mechanism.

When you implement `curam.core.hook.impl.CaseClientsPopulationHook`, you can customize the clients that are returned and displayed according to one of the following options:

- The `CaseParticipantRoles` of type 'Member' or 'Primary Client'
- The members of the `CaseGroups` of type 'Member' that are associated with the case, for product delivery cases that have been customized to make use of case groups.

By default, the `CaseParticipantRoles` of type 'Member' or 'Primary Client' are returned for all case types.

Configuring special cautions

Agencies can define special cautions through the `Special Caution Category` and `Special Caution Type` code tables. By default, multiple special cautions of the same type are prevented from overlapping for the same period. You can enable special cautions to overlap by disabling some validations.

About this task

Hierarchically, the `Special Caution Category` code table is a parent of the `Special Caution Type` code table. To add a special caution type to a category, add an item to the `Special Caution Type` code table, then add the category parent code to the new special caution type in the `Special Caution Category` code table.

To enable special cautions of the same type to overlap for the same period, disable the following validations:

- `specialcaution.err_specialcaution_xrv_category_and_type_overlaps_in_specified_date_range|a|`
- `specialcaution.err_specialcaution_xrv_category_and_type_overlaps_in_specified_date_range|a|1`

- `specialcaution.err_specialcaution_xrv_category_code_and_type_code_exist|a|`
- `specialcaution.err_specialcaution_xrv_category_code_and_type_code_exist|a|1`

Procedure

Create a special caution type in the `Special Caution Type` code table.

1. Log on to the IBM Cúram Social Program Management application as a system administrator, and click **System Configurations**.
2. In the **Shortcuts** menu, click **Application Data** > **Code Tables**.
3. Search for `Special Caution Type`.
4. Select **New Item** from the `Special Caution Type` list actions menu.
5. Enter a name and a technical ID for the new special caution type, where the technical ID can be set to any unique string.
6. Save and publish the update.

Define the hierarchy between the new special caution type and the parent special caution category.

7. In the **Shortcuts** menu, click **Application Data** > **Code Table Hierarchies**, and then click **SpecialCautionHierarchy**.
8. Under **Code Tables**, search for and expand **Special Caution Type**.
9. Locate the new special caution type in the list and select **Change parent code** from the associated list action menu.
10. To link the special caution type to the appropriate category in the `Special Caution Category` code table, select the appropriate parent code from the list.
11. Save and publish the update.

Configuring the display financials date range

Agencies can use the Display Financials Date Range **`curam.financial.generatefinancialsdaterange`** application property to determine the period that is covered by the case and participant financial searches. The default value is 365.

The case and participant financial transaction searches are used to retrieve all financial instructions from a date in the past that is determined by the latest financial transaction effective date, less the number of days that are specified by the **`curam.financial.generatefinancialsdaterange`** application property.

Implementing rate tables

Rate tables are used to maintain values that can vary over time. An example of a rate is a Gross Income limit based on the number of people in a household.

Configuring rate tables

Rate tables can be created and maintained independently, which allows for a more flexible approach to rates that apply to products.

Rate tables can be used for values that are effective during set time periods. For example, the reassessment of a case might call upon values in the past, where the rates have changed considerably, and these values can be stored on a rate table.

A rate table can contain an indefinite number of rows and columns that determine the number of cells. The values of individual rates are stored in these cells. Also, subrows and subcolumns can be added to each individual row and column. A rate table also can be cloned and used as a basis for a new rate table.

Each rate table has a type and effective date.

Note: If two rate tables are created for the same type, then the effective date of the rate tables continues until midnight of the day before the effective date of the next rate table.

Simple and complex rate tables

Rate tables are designed to store any value or set of values that can vary over time. Rate tables can be simple or complex.

A simple rate table might have a single column with a single row that contains a value. Complex rate tables can consist of multiple columns that contain multiple subcolumns and multiple rows. In addition to a value, each cell can contain a maximum and a minimum value. The interpretation of the values in a rate table depends on the context in which they are used.

Rate tables can be described as follows:

- Rates are stored in table structures.
- You can maintain rates independent from either rules or code.
- Rates are inherently effective date based. Rates are continually raised and lowered in line with budgetary changes. Each set of rates should be effective until a later set of effective dated rates is entered.
- Rate tables can be one or two dimensional with a type, and minimum and maximum values associated with each entry. (Effectively resulting in three dimensional tables).
- Users can maintain rate tables through the administration application.
- Business processes access rate table through code.

Examples of rate tables include:

- Age thresholds, for example under 65, over 18.
- Monetary limits, for example earnings must be at least $4,000, capital must be less than $10,000.
- Limits on the numbers of hours worked per week.
- Tax rates to be applied across a number of benefits.

Rate table entities

The `RateHeader`,`RateColumnRateCell` and `RateRow` entitles are used in rate tables.

Rate header

The rate header entity stores header information for a rate table. Each record on this table refers to a rate table with an effective date and a status. It contains the following fields:

Table 6. Summary of Rate Header Entity Fields

Entity Field	Description
Effective Date	This field is the date on which this record becomes effective.
Rate Table Type	This field is the type of the rate table. It is populated from the `RateTableType` code table. There should be one entry in this code table for each rate table in the system. This field is used to determine the set of temporal records that comprise the data for a given rate table.

Table 6. Summary of Rate Header Entity Fields (continued)	
Entity Field	**Description**
Rate Status	This field is the status of this record. It is populated from the `RateStatus` code table. Where a second record has been cloned from a rate table the original record will have a rate status of superseded.
Comments	This field contains the comments about this record.

Rate column

The rate column entity represents a column in a given rate table. Each column can contain one or more cells. It contains the following fields:

Table 7. Summary of Rate Column Entity Fields	
Entity Field	**Description**
Rate Header ID	This field is the ID of the `RateHeader` with which this record is associated.
Parent Column ID	This field is the ID of the parent column of this column. This will only be populated for sub-columns. This field is optional.
Rate Column Key Min	This field is the minimum value for this column. This field is optional.
Rate Column Key Max	This field is the maximum value for this column. This field is optional.
Rate Column Key Value	This field is the value for this column. When there is only a single row in a table, this value is used. This field is optional.
Rate Column Type	This field is the type of the column. It is populated from the `Rate ColumnType` code table. There should be one entry in this code table for each rate column in the system.
Rate Column Index	This field is the index of the column within the rate table. If the column has no parent, this field refers to the index of the column within the table as a whole. If the column has a parent (making it a sub-column), then this field refers to the index of the column within the parent column.
Any Minimum Indicator	This field is used to indicate if any minimum value has been set for this column. If the value of this field is 'False', then no minimum value has been set for this column.
Any Maximum Indicator	This field is used to indicate if any maximum value has been set for this column. If the value of this field is `False`, then no maximum value has been set for this column.

Rate row

The rate row entity represents a row in a rate table. Each rate row record can relate to one or more rate cell records. It contains the following fields:

Table 8. Summary of Rate Row Entity Fields

Entity Field	Description
Rate Header ID	This field is the ID of the related rate header record.
Rate Row Type	This field is the type for this row. This field is optional.
Rate Row Index	This field is the index of the row within the rate table

Rate cell

The rate cell entity represents a cell in the rate table. It contains the following fields:

Table 9. Summary of Rate Cell Entity Fields

Entity Field	Description
Rate Column ID	This field is the ID of the `RateColumn` record with which the cell is associated.
Rate Cell Min	This field is the minimum value for this cell. If a minimum value is entered, a maximum value must be entered. Also, the minimum value cannot be greater than the maximum value.
Rate Cell Max	This field is the maximum value for this cell. If a maximum value is entered, a minimum value must be entered.
Rate Cell Value	This field is the value for this cell. This value is mandatory if no minimum and maximum value are entered.
Rate Cell Index	This field is the index (row) of the cell within the column.
Any Minimum Indicator	This field is used to indicate if any minimum value has been set for this cell. If the value of this field is 'True', then no minimum value has been set for this cell.
Any Maximum Indicator	This field is used to indicate if any maximum value has been set for this cell. If the value of this field is 'True', then no maximum value has been set for this cell.
Rate Row ID	This field is the ID of the *RateRow* record with which the cell is associated.
Blank Cell Indicator	This field is used to indicate whether the rate *RateCell* has any value recorded.

Rate table business processes

The rate table business process contains a number of methods that are used to maintain rate tables in the application. A separate business process, `ParseRates`, is used to parse rate table records into the XML format used to drive the display of rate tables.

`RateTable` methods

A list of the methods supplied by the `RateTable` class.

Clone Rate Table

This method clones a rate table. The cloned rate table has the type and effective date specified as parameters.

Create Rate Cell

This method creates a rate cell. It adds a rate cell record to the database.

Create Rate Column

This method creates a rate column. It adds a rate column record to the database.

Create Rate Table Header

This method creates a rate table header. It adds a rate header record to the database.

Create SubColumn

This method creates a rate column. It adds a rate column record to the database that relates though a `parentColumnID` to the parent column.

Create SubRow

This method creates a subrow. It adds a rate row record to the database that relates though a `parentRowID` to the parent row.

List Rate Table

This method returns a list of available rate tables.

List Rate Table History

This method returns a list of rate header records by rate table type (all), effective date, or status.

Modify Rate Cell

This method modifies rate cell data.

Modify Rate Column

This method modifies rate column data.

Modify Rate Header

This method modifies rate header data.

Modify Rate Row

This method modifies rate row data.

Modify SubColumn

This method modifies subcolumn data.

Read Rate Cell Data

This method reads the details of a rate cell.

Read Rate Column

This method reads the details of a rate column.

Read Rate Header

This method reads the details of a rate header.

Read Rate Row

This method reads the details of a rate row.

Read Rate Table

This method reads the details of the rate table from the database. It performs a Rate Header read and then calls the `encodeRateTableData` method on the `ParseRates` class to get the remaining details.

Remove Rate Cell

This method removes a cell from a rate table.

Remove Rate Column

This method removes a column from a rate table.

Remove Rate Row

This method removes a row from a rate table.

Remove SubColumn

This method removes a subcolumn from a rate table.

Remove SubRow

This method removes a subrow from a rate table.

`ParseRates` methods

A list of the methods supplied by the `ParseRates` class.

Encode Rate Table Date

This method reads the rate data from the database and encodes it into the XML format. For more information, see "Rate table XML format" on page 50.

Update Indexes For Rate Cell

This method sets the `RateCell` record index after any Rate Table record modifications that require reindexing for the cell.

Update Indexes For Rate Column

This method sets the `RateColumn` record index after any Rate Table record modifications that require reindexing for the column.

Update Indexes For Rate Row

This method sets the `RateRow` record index after any Rate Table record modifications that require reindexing for the row.

Update Indexes For Sub-Column

This method sets the subcolumn `RateColumn` record index after any Rate Table record modifications that require reindexing for the subcolumn.

Update Indexes For Sub-Row

This method sets the `RateCell` record index after any Rate Table record modifications that require reindexing for the subrow.

Rate table XML format

The XML format of the rate table data, which is interpreted by the client to display the rate table.

RATES_DATA

Each rate table description begins with this tag. This tag has a READ_ONLY attribute. It contains a number of columns.

COLUMN

This tag has INDEX, ID, and TYPE attributes. Each column can contain a number of rows or sub-columns.

ROW

This tag has INDEX, ID, TYPE, and NUM_SUB_ROWS attributes. Each row can contain a number of cells.

SUB_COLUMN

This tag has INDEX, ID, MAXIMUM, MINIMUM, and VALUE attributes. Each column can contain a number of rows.

CELL

This tag has ID, VALUE, MAXIMUM, MINIMUM, and COLUMN_INDEX attributes.

Example usage of rate tables

Possible uses of rate tables include two-dimensional rate tables, two-dimensional rate tables with ranges, and two-dimensional rate tables with ranges and subcolumns.

Two-dimensional rate table

- Any number of columns, each with a type.
- Any number of rows, with corresponding number of cells.

Table 10. Example two-dimensional rate table

	New Employer (%)	**Q Rate Employer (%)**	**G Rate Employer (%)**	**F Rate Employer (%)**
Normal Rate	1	2	4	8
Subsidiary Rate	2	4	8	16
RESF Rate	0.075	0.075	0.075	0.075

The XML string for this rate table is shown. CT1 implies New Employer %, RT1 implies Normal Rate, and so on.

```
<RATES_DATA>
<COLUMN  INDEX=1  ID=1  TYPE=CT1>
<ROW INDEX=1  ID =2  TYPE=RT1>
<CELL ID=3  VALUE=1  COLUMN_INDEX=1/>
</ROW>
<ROW INDEX=2  ID =4  TYPE=RT2>
<CELL ID=5  VALUE=2  COLUMN_INDEX=2/>
</ROW>
<ROW INDEX=3  ID =6  TYPE=RT3>
<CELL ID=7  VALUE=0.075  COLUMN_INDEX=3/>
</ROW>
</COLUMN>
<COLUMN  INDEX=2  ID=8  TYPE=CT2>
<ROW INDEX=1  ID =2  TYPE=RT1>
<CELL ID=10  VALUE=2  COLUMN_INDEX=1/>
```

```
</ROW>
<ROW INDEX=2  ID =4  TYPE=RT2>
<CELL ID=12  VALUE=4  COLUMN_INDEX=2/>
</ROW>
<ROW INDEX=3  ID =6  TYPE=RT3>
<CELL ID=14  VALUE=0.075  COLUMN_INDEX=3/>
</ROW>
</COLUMN>
<COLUMN  INDEX=3  ID=15  TYPE=CT3>
<ROW INDEX=1  ID =2  TYPE=RT1>
<CELL ID=17  VALUE=8  COLUMN_INDEX=1/>
</ROW>
<ROW INDEX=2  ID =4  TYPE=RT2>
<CELL ID=19  VALUE=16  COLUMN_INDEX=2/>
</ROW>
<ROW INDEX=3  ID =6  TYPE=RT3>
<CELL ID=21  VALUE=0.075  COLUMN_INDEX=3/>
</ROW>
</COLUMN>
<COLUMN  INDEX=4  ID=22  TYPE=CTS4>
<ROW INDEX=1  ID =2  TYPE=RT1>
<CELL ID=24  VALUE=4  COLUMN_INDEX=1/>
</ROW>
<ROW INDEX=2  ID =4  TYPE=RT2>
<CELL ID=26  VALUE=8  COLUMN_INDEX=2/>
</ROW>
<ROW INDEX=3  ID =6  TYPE=RT3>
<CELL ID=28  VALUE=0.075  COLUMN_INDEX=3/>
</ROW>
</COLUMN>
</RATES_DATA>
```

Two-dimensional rate table with ranges

A two-dimensional rate table with ranges consists of:

- Any number of columns, each with a type.
- Any number of rows, each with a type.
- Any number of cells, each with:
  - A minimum value
  - A maximum value
  - A value

Table 11. Example two-dimensional rate table with ranges

	Account Percentage (%)	**New Employer (%)**	**Q Rate Employer (%)**	**G Rate Employer (%)**
Normal Rate	0 : 1.5 1.6 : 3 3	1 1.5 1.6	2 2.1 2.2	4 4.1 4.2
Subsidiary Rate	0 : 1.5 1.6 : 3 3	2 2 2	4 4 4	8 8 8
RESF Rate	0 : 100	0.075	0.075	0.075

This example illustrates the format that is used by the Cúram Reference Application to display minimum and maximum values for a cell. The minimum and maximum values for a cell are separated with a colon (':'). For example, '0 : 1.5'. If a value is specified in addition to a range, the value is separated from the range with another colon character. For example, '0 : 1.5 : 4'.

In the example, the account percentage column is used to determine the rate appropriate for each different employer type. For example, a Q Rate Employer with an account percentage value of 1.9%, is charged 2.1% for the Normal Rate and 4% for the Subsidiary Rate.

The XML string for this rate table is shown. CT1 implies Account Percentage %, RT1 implies Normal Rate, and so on.

```
<RATES_DATA>
<COLUMN  INDEX=1  ID=1  TYPE=CT1>
<ROW INDEX=1  ID =2  TYPE=RT1 NUM_SUB_ROWS=3>
<CELL ID=3 MINIMUM=0.0 MAXIMUM=1.5 COLUMN_INDEX=1/>
<CELL ID=4 MINIMUM=1.6 MAXIMUM=3.0 COLUMN_INDEX=2/>
<CELLID=5 VALUE=3.0  COLUMN_INDEX=3/>
</ROW>
<ROW INDEX=2  ID =6  TYPE=RT2 NUM_SUB_ROWS=3>
<CELL ID=7 MINIMUM=0.0 MAXIMUM=1.5 COLUMN_INDEX=1/>
<CELL ID=8 MINIMUM=1.6 MAXIMUM=3.0 COLUMN_INDEX=2/>
<CELL ID=9 VALUE=3.0  COLUMN_INDEX=3/>
</ROW>
<ROW INDEX=3  ID =10  TYPE=RT3>
<CELL ID=11 MINIMUM=0.0 MAXIMUM=100.0 COLUMN_INDEX=3/>
</ROW>
</COLUMN>
<COLUMN  INDEX=2  ID=12 TYPE=CT2>
<ROW INDEX=1  ID =2  TYPE=RT1 NUM_SUB_ROWS=3>
<CELL ID=13  VALUE=1  COLUMN_INDEX=1/>
<CELL ID=14  VALUE=1.5  COLUMN_INDEX=2/>
<CELL ID=15  VALUE=1.6  COLUMN_INDEX=3/>
</ROW>
<ROW INDEX=2  ID =6  TYPE=RT2 NUM_SUB_ROWS=3>>
<CELL ID=16  VALUE=2  COLUMN_INDEX=1/>
<CELL ID=17  VALUE=2  COLUMN_INDEX=2/>
<CELL ID=18  VALUE=2  COLUMN_INDEX=3/>
</ROW>
<ROW INDEX=3  ID =10  TYPE=RT3>
<CELL ID=19  VALUE=0.075  COLUMN_INDEX=3/>
</ROW>
</COLUMN>
<COLUMN  INDEX=3  ID=20 TYPE=CT3>
<ROW INDEX=1  ID =2  TYPE=RT1 NUM_SUB_ROWS=3>
<CELL ID=21  VALUE=2  COLUMN_INDEX=1/>
<CELL ID=22  VALUE=2.1  COLUMN_INDEX=2/>
<CELL ID=23  VALUE=2.2  COLUMN_INDEX=3/>
</ROW>
<ROW INDEX=2  ID =6  TYPE=RT2 NUM_SUB_ROWS=3>>
<CELL ID=24  VALUE=4  COLUMN_INDEX=1/>
<CELL ID=25  VALUE=4  COLUMN_INDEX=2/>
<CELL ID=26  VALUE=4  COLUMN_INDEX=3/>
</ROW>
<ROW INDEX=3  ID =10  TYPE=RT3>
CELL ID=27 VALUE=0.075  COLUMN_INDEX=3/>
</ROW>
</COLUMN>
</COLUMN>
<COLUMN  INDEX=4  ID=28 TYPE=CT4>
<ROW INDEX=1  ID =2  TYPE=RT1 NUM_SUB_ROWS=3>
<CELL ID=29  VALUE=4  COLUMN_INDEX=1/>
<CELL ID=30  VALUE=4.1  COLUMN_INDEX=2/>
<CELL ID=31  VALUE=4.2  COLUMN_INDEX=3/>
</ROW>
<ROW INDEX=2  ID =6  TYPE=RT2 NUM_SUB_ROWS=3>>
<CELL ID=32  VALUE=8  COLUMN_INDEX=1/>
<CELL ID=33  VALUE=8  COLUMN_INDEX=2/>
<CELL ID=34  VALUE=8  COLUMN_INDEX=3/>
</ROW>
<ROW INDEX=3  ID =10  TYPE=RT3>
<CELL ID=35 VALUE=0.075  COLUMN_INDEX=3/>
</ROW>
</COLUMN>
</RATES_DATA>
```

Two-dimensional rate table with ranges and subcolumns

A two-dimensional rate table with ranges and sub columns consists of:

- Any number of columns or subcolumns, each with:

  - A type
  - A minimum value
  - A maximum value
- Any number of rows each with:
  - A type
- Any number of cells each with:
  - A minimum value
  - A maximum value
  - A value

Table 12. Example two-dimensional rate table with ranges and subcolumns

	Account Percentage	**New Employer (%)**	**Q Rate (%)**	**G Rate Employer (%)**
Size of Fund Index		0 : 1001 10000 20000	0 : 1001 10000 20000	0 : 1001 10000 20000
Normal Rate	0:1.5 1:6.3 3	1 1.1 1.5 1.6 1.6 1.7	2 2.1 2.1 2.2 2.2 2.3	2 4 2.1 4.1 2.2 4.2
Normal Rate	0:1.5 1:6.3 3	2 2.1 4 4.1 8 8.1	2 2.2 4.1 4.2 8.1 8.2	2.2 2.3 4.2 4.3 8.2 8.3
RESF Rate	0 : 100	0.075 0.075	0.075 0.075	0.075 0.075

In the example, the size of fund index values are used in addition to the account percentage value to determine the rate for each employer type. For example, a Q Rate Employer with an account percentage value of 1.9% and a fund index of 9000 is charged 2.1% for the Normal Rate and 4.1% for the Subsidiary Rate.

The XML string for this rate table is shown. CT1 implies Account Percentage %, RT1 implies Size of Fund Index, and so on.

```
<RATES_DATA>
<COLUMN  INDEX=1  ID=1  TYPE=CT1>
<ROW
INDEX=1  ID =2  TYPE=RT1 NUM_SUB_ROWS=3>
<CELL
ID=3 MINIMUM=0.0 MAXIMUM=1.5 COLUMN_INDEX=1/>
<CELL
ID=4 MINIMUM=1.6 MAXIMUM=3.0 COLUMN_INDEX=2/>
<CELL
ID=5 VALUE=3.0  COLUMN_INDEX=3/>
</ROW>
<ROW INDEX=2  ID =6  TYPE=RT2 NUM_SUB_ROWS=3>
<CELL ID=7 MINIMUM=0.0 MAXIMUM=1.5 COLUMN_INDEX=1/>
<CELL ID=8 MINIMUM=1.6 MAXIMUM=3.0 COLUMN_INDEX=2/>
<CELL ID=9 VALUE=3.0  COLUMN_INDEX=3/>
</ROW>
<ROW INDEX=3  ID =10  TYPE=RT3>
<CELL ID=11
MINIMUM=0.0 MAXIMUM=100.0 COLUMN_INDEX=3/>
</ROW>
```

```
</COLUMN>
<COLUMN  INDEX=2  ID=12
TYPE=CT2>
<SUB_COLUMN INDEX=1 ID=13 MAXIMUM=10000.0
MINIMUM=0.0>
<ROW INDEX=1  ID =2  TYPE=RT1 NUM_SUB_ROWS=3>
<CELL ID=14  VALUE=1  COLUMN_INDEX=1/>
<CELL ID=15
 VALUE=1.5  COLUMN_INDEX=2/>
<CELL ID=16  VALUE=1.6
 COLUMN_INDEX=3/>
</ROW>
<ROW
INDEX=2  ID =6  TYPE=RT2 NUM_SUB_ROWS=3>>
<CELL
ID=17  VALUE=2  COLUMN_INDEX=1/>
<CELL ID=18
 VALUE=2.1  COLUMN_INDEX=2/>
<CELL ID=19  VALUE=2.2
 COLUMN_INDEX=3/>
</ROW>
<ROW
INDEX=3  ID =10  TYPE=RT3>
<CELL ID=20  VALUE=0.075
 COLUMN_INDEX=3/>
</ROW>
</SUBCOLUMN>
<SUB_COLUMN INDEX=2 ID=21 MAXIMUM=20000.0 MINIMUM=10001.0>
<ROW INDEX=1  ID =2  TYPE=RT1 NUM_SUB_ROWS=3>
<CELL ID=22  VALUE=1.1  COLUMN_INDEX=1/>
<CELL ID=23  VALUE=1.6  COLUMN_INDEX=2/>
<CELL
ID=24  VALUE=1.7  COLUMN_INDEX=3/>
</ROW>
<ROW INDEX=2  ID =6  TYPE=RT2 NUM_SUB_ROWS=3>>
<CELL ID=25  VALUE=2.1  COLUMN_INDEX=1/>
<CELL
ID=26  VALUE=2.2  COLUMN_INDEX=2/>
<CELL ID=27
 VALUE=2.3  COLUMN_INDEX=3/>
</ROW>
<ROW INDEX=3  ID =10  TYPE=RT3>
<CELL
ID=28  VALUE=0.075  COLUMN_INDEX=3/>
</ROW>
</SUBCOLUMN>
</COLUMN>
<COLUMN  INDEX=3  ID=29 TYPE=CT3>
<SUB_COLUMN
INDEX=1 ID=30 MAXIMUM=10000.0 MINIMUM=0.0>
<ROW
INDEX=1  ID =2  TYPE=RT1 NUM_SUB_ROWS=3>
<CELL ID=31
 VALUE=2  COLUMN_INDEX=1/>
<CELL ID=33  VALUE=2.1
 COLUMN_INDEX=2/>
<CELL ID=33  VALUE=2.2  COLUMN_INDEX=3/>
</ROW>
<ROW INDEX=2  ID =6  TYPE=RT2
NUM_SUB_ROWS=3>>
<CELL ID=34  VALUE=4  COLUMN_INDEX=1/>
<CELL ID=35  VALUE=4.1  COLUMN_INDEX=2/>
<CELL
ID=36  VALUE=4.2  COLUMN_INDEX=3/>
</ROW>
<ROW INDEX=3  ID =10  TYPE=RT3>
<CELL
ID=37 VALUE=0.075  COLUMN_INDEX=3/>
</ROW>
</SUBCOLUMN>
<SUB_COLUMN INDEX=2
ID=38 MAXIMUM=20000.0 MINIMUM=10001.0>
<ROW
INDEX=1  ID =2  TYPE=RT1 NUM_SUB_ROWS=3>
<CELL
ID=39  VALUE=2.1  COLUMN_INDEX=1/>
<CELL ID=40
 VALUE=2.2  COLUMN_INDEX=2/>
<CELL ID=41  VALUE=2.3
 COLUMN_INDEX=3/>
</ROW>
<ROW
```

```
INDEX=2  ID =6  TYPE=RT2 NUM_SUB_ROWS=3>>
<CELL
ID=42  VALUE=4.1  COLUMN_INDEX=1/>
<CELL ID=43
 VALUE=4.2  COLUMN_INDEX=2/>
<CELL ID=44  VALUE=4.3
 COLUMN_INDEX=3/>
</ROW>
<ROW
INDEX=3  ID =10  TYPE=RT3>
<CELL ID=45 VALUE=0.075
 COLUMN_INDEX=3/>
</ROW>
</SUBCOLUMN>
</COLUMN>
</COLUMN>
<COLUMN
 INDEX=4  ID=46 TYPE=CT4>
<SUB_COLUMN INDEX=1
ID=47 MAXIMUM=10000.0 MINIMUM=0.0>
<ROW INDEX=1
 ID =2  TYPE=RT1 NUM_SUB_ROWS=3>
<CELL ID=48
 VALUE=2  COLUMN_INDEX=1/>
<CELL ID=49  VALUE=2.1
 COLUMN_INDEX=2/>
<CELL ID=50  VALUE=2.2  COLUMN_INDEX=3/>
</ROW>
<ROW INDEX=2  ID =6  TYPE=RT2 NUM_SUB_ROWS=3>>
<CELL ID=51  VALUE=8  COLUMN_INDEX=1/>
<CELL ID=52
 VALUE=8.1  COLUMN_INDEX=2/>
<CELL ID=53  VALUE=8.2
 COLUMN_INDEX=3/>
</ROW>
<ROW
INDEX=3  ID =10  TYPE=RT3>
<CELL ID=54 VALUE=0.075
 COLUMN_INDEX=3/>
</ROW>
</SUBCOLUMN>
<SUB_COLUMN INDEX=2 ID=55 MAXIMUM=20000.0 MINIMUM=10001.0>
<ROW INDEX=1  ID =2  TYPE=RT1 NUM_SUB_ROWS=3>
<CELL ID=56  VALUE=4  COLUMN_INDEX=1/>
<CELL ID=57
 VALUE=4.1  COLUMN_INDEX=2/>
<CELL ID=58  VALUE=4.2
 COLUMN_INDEX=3/>
</ROW>
<ROW
INDEX=2  ID =6  TYPE=RT2 NUM_SUB_ROWS=3>>
<CELL
ID=59  VALUE=8.1  COLUMN_INDEX=1/>
<CELL ID=60
 VALUE=8.2  COLUMN_INDEX=2/>
<CELL ID=61  VALUE=8.3
 COLUMN_INDEX=3/>
</ROW>
<ROW
INDEX=3  ID =10  TYPE=RT3>
<CELL ID=62 VALUE=0.075
 COLUMN_INDEX=3/>
</ROW>
</SUBCOLUMN>
</COLUMN>
</RATES_DATA>
```

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 10504-1785 US

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, Nihonbashi-Hakozakicho, Chuo-ku Tokyo 103-8510, Japan

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:

IBM Director of Licensing IBM Corporation North Castle Drive, MD-NC119 Armonk, NY 10504-1785 US

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBMproducts. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

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.

The registered trademark Linux is used pursuant to a sublicense from the Linux Foundation, the exclusive licensee of Linus Torvalds, owner of the mark on a worldwide basis.

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.

Part Number:

(1P) P/N: