



# **Cúram 8.2.2**

## **Provider Management Configuration Guide**



## Note

---

Before using this information and the product it supports, read the information in [Notices on page 29](#)



# Edition

---

This edition applies to Cúram 8.2.2.

© Merative US L.P. 2012, 2026

Merative and the Merative Logo are trademarks of Merative US L.P. in the United States and other countries.



# Contents

---

<b>Note.....</b>	<b>iii</b>
<b>Edition.....</b>	<b>v</b>
<b>1 Configuring Provider Management.....</b>	<b>9</b>
1.1 Provider Administration.....	9
Securing CPM External Access.....	9
Defining the Email Server for Providers and Provider Groups.....	10
Configuring the Renewal Period for Provider Contracts.....	10
Setting up the Renewal Period for Provider Licenses.....	10
Setting Up a New Provider Training Course.....	10
Configuring Training Requirements for Licenses.....	11
Configuring Certifications for Provider Members.....	11
1.2 Service Offering Administration.....	12
Registering Service Offerings.....	12
Registering Training Service Offerings.....	14
Configuring Service Offering Financials.....	14
Grouping Service Offerings.....	15
Searching and Viewing Service Offerings.....	15
Defining Approval Criteria for Service Offerings.....	15
Preventing Incidental Authorization of Service Offerings.....	16
Keywords.....	16
Evidence.....	16
Setting Up Attendance Tracking for Service Offerings.....	16
Attendance Configuration Options.....	18
Recording Absence Reasons for Service Offerings.....	18
Indicating if Attendance Is Required for Payment.....	18
1.3 Service Evaluation Configuration.....	18
Configuring Evaluation Criteria.....	19
1.4 CPM Batch Processes.....	19
Batch Parameters and Processing Date.....	19
RosterSubmissionDue.....	20
ProcessContractRenewalNotifications.....	20
ProcessLicenseRenewalNotifications.....	20
ActivateCPMLiabilityCases.....	21
CreatePlacementEvidence.....	21
ExpireReservation.....	21
GenerateCPMPayslips.....	22
GenerateRosters.....	22
RosterSubmissionReminder.....	22

- ActivateProviderPaymentCorrectionCases..... 23
- ImportTaxonomyData.....23
- (deprecated) PublishTaxonomyData..... 24
- TaxonomyDatabaseSearchExtractor..... 24
- 1.5 Summary of CPM Application Properties.....24
  - External User Security Settings..... 24
  - Application – CPM Attendance Settings..... 25
  - Application – CPM Financials Settings..... 25
  - Application – CPM Facility Settings..... 26
  - Application - CPM Contracts Settings.....27
  - Application CPM License Settings..... 27
  - Application - Sequential Continuous Reference Numbers..... 27
  - Application - Miscellaneous Properties..... 28
  - Infrastructure - Miscellaneous Properties.....28
  - Case Parameters.....28
  - Application - CPM Dynamic SQL Enabled.....28

- Notices..... 29**
  - Privacy policy..... 30
  - Trademarks..... 30

# 1 Configuring Provider Management

---

You can configure provider trainings and certifications. Provider service offerings can be configured including information on service offerings, registration, financial settings, and attendance tracking. There are a number of Provider Management batch processes and application properties that can be configured.

## Purpose

The following pages describe the configuration options that are available for Provider Management (CPM). For providers, this includes information on configuring trainings and certifications. For service offerings, this includes information on service offering registration, financial settings and attendance tracking.

## Audience

The target audience is administrators responsible for the configuration of CPM. It is assumed that the CPM administrator has worked with code tables, application properties, and system security as part of system administration. These configuration options are managed by an administrator in the runtime administration application and the runtime system administration application. There is no software development required to make these configuration changes.

## Prerequisites

Users must be familiar with the business concepts of CPM. For more information, see the *Provider Management Guide* related link. For administrators and developers who customize CPM, see *Developing with Provider Management*.

## Related information

## 1.1 Provider Administration

---

The provider configuration options, that can be set in the administration and system administration applications, permit the resource manager to manage individual providers in the resource manager application, according to the organization's requirements.

Information is provided on configuration options for external provider access, grace periods, contracts, and licenses. The process of setting up a new provider training is discussed. Finally, the configuration of provider certifications and license training requirements is covered.

## Securing CPM External Access

The property `curam.custom.externalaccess.implementation` is used to secure the external application. This property specifies the CPM specific implementation file for external access security. The default value is `curam.cpm.eua.facade.impl.CPMExternalAccessSecurity`. This class is used to authenticate external users and authorize them to use specific parts of the application. In

the external provider application, for example, the external provider user can view the details of a provider offering but cannot modify or cancel provider offerings.

If external access for providers as outlined above is not specified, then the external access will be limited by the `curam.core.sl.impl.PublicAccessSecurity` value for the above property. In this case, the external provider will be able to view their participant details only. For more information on securing the external application, please consult the *Cúram Server Developer Guide*.

## Defining the Email Server for Providers and Provider Groups

The property `curam.cpmprovider.cpmprovideremailserver` is used to define the email server used for sending and receiving emails to and from providers and provider groups. It is important for customers to set their own value for the email server. Out of the box, the default value is `smtp.curamsoftware.com`.

## Configuring the Renewal Period for Provider Contracts

The property `curam.contracts.renewal.noofdaysevents` is used to set the renewal period in days, for all the contracts within the organization. When this property is set, the resource manager will be issued with an automatic notification that the contract is due to expire and the provider will be notified via their preferred communication method. For example, Linda Smith, a child care provider provider, has a 6 month contract with the organization. 7 days before Linda's contract is due to expire she is sent an automatic email by the system to tell her about her imminent contract expiration. The resource manager responsible for managing Linda Smith is also notified.

The default value for this property is 7 days.

## Setting up the Renewal Period for Provider Licenses

The property `curam.cpm.license.renewal.noofdaysevents` is used to set the renewal period in days, for all the licenses within the organization. A notification is issued to the responsible resource manager a set number of days before a license is due to be renewed. The provider in question will also be notified via their preferred communication method.

The default value is 7 days.

## Setting Up a New Provider Training Course

When a training course is recorded, a training type must be specified. This allows the training course to be categorized, making it easier for the resource manager to search for and select the appropriate training when adding a training to a provider's profile. For example, a category of Health and Safety could describe a number of different trainings: First Aid, Workplace Safety, etc.

Once the training course is recorded in the administration application, resource managers can then add the training to a provider's record. At this point, any provider members that are to undergo the training can be selected. Once a training has been created in the system, administrators can then add credits to the training record.

## ***Adding Training Credits***

Training credits allow the CPM administrator to customize how provider members receive and are awarded for trainings that have been undertaken by provider members. A provider type and category can be specified. If no category and type are selected for the training credit, then the value entered will be treated as a default.

To illustrate: both a foster care provider member and a child care provider member may require a training program in early years child care in order to provide their services. However, the relative importance of the course may be different between the two provider members. The administrator can specify two different types of training credit for the early years child care training, one for the foster care provider member, and one for the child care provider member.

**Note:** Only one training credit per training can be specified for any one provider category and type combination. This allows the organization to ensure that all provider members in a certain provider category and type meet specific training standards.

## **Configuring Training Requirements for Licenses**

Administrators can specify training requirements for provider licenses. Multiple training requirements can be added for each license type that is recorded in the system. If a training requirement is specified for a license, provider members may or may not need to have completed the training in order to be licensed by the organization, depending on whether the training is required or recommended to be completed.

For example, a provider Safe Care Ltd works with the organization to provide child care to the organization's clients. To ensure that Safe Care Ltd's employees provide a high standard of care, a child development training course is specified as a requirement in order for the provider to be licensed as a child care agency. The training is specified as required to be completed by Safe Care Ltd provider members, as it is essential that provider members are properly licensed when working with children. An administrator selects the license type of Child Care and adds the training as a requirement for the license type.

## **Configuring Certifications for Provider Members**

A certification is an official recognition of a provider member's skill or qualification. As such, certifications are applied to individual provider members. They are not applied at the provider level. When configuring a certification, the credits required to achieve the certification and the issuing body can be specified. Certification credits allow the organization to configure what is required of the provider member before they can be certified. After the certification is recorded in the administration application, resource managers can apply it to individual provider members.

For example, a CPR Practitioner certification is required by every provider member working with a provider, Senior Care Ltd., a residential care facility. An administrator creates a certification of type CPR Practitioner with a validity period of one year. Once the certification is recorded in the administration application, the resource manager can then record the certification for Senior Care provider members.

## 1.2 Service Offering Administration

---

The configuring service offerings include grouping service offerings and setting approval criteria for service offerings.

In addition, information is provided on setting service offerings to be paid based on attendance or placement, configuring attendance settings for service offerings, and recording absence reasons for service offerings.

### Registering Service Offerings

All service offerings are registered in the organization's service registry. A unit of measure must be indicated for all service offerings. The unit of measure will depend on the type of service offering that is being recorded. For example, residential placement services should have a unit of measure of place. The unit of measure for non-placement service offerings include, but are not limited to, doses, treatments, and visitations.

Administrators can additionally specify whether the service offering is of a special condition type, where service offerings have other attributes that would necessitate them being treated differently in some way. For example, special conditions can be used to differentiate between secure and non secure types of service offerings.

Once a service offering is recorded in the administration application, resource managers can then associate the newly recorded service offering with any provider who offers the service to the organization's clients.

### ***Configuring Maximum Units and Frequency for a Service Offering***

The maximum units and maximum units frequency for a service offering can be configured by an administrator. These settings are used to control the number of units of a service a client can receive in a given period. For example, Jane Smith is authorized to receive 30 units of physiotherapy. The physiotherapy provider offering has a maximum units of 5 and a maximum units frequency of one week. A provider, Mark's Gym, is contracted to provide a maximum of 5 units of physiotherapy per week.

### ***Configuring Service Rates***

A service rate is the amount the organization is willing to pay for a unit of a service offering, for example, the amount to be paid for a placement or for a training session. Default service rates can be configured for service offerings by administrators. This can be a fixed amount or an amount which falls within a range (between a minimum and maximum rate) for a defined period of time.

A number of service rates can be maintained for a service offering, however, only one active service offering rate can be specified for a service offering for a given time period. Resource managers can also specify service rates for individual service offerings at the provider level in the resource manager application. The payment options for a service offering configured by an administrator will apply for any service rates set by the resource manager.

Minimum, maximum, and fixed rates dictate the bounds between which an invoiced unit of a service will be paid. A service can be paid for at a fixed rate per unit (with some flexibility

provided by the payment options), or at between a range of acceptable rates which can be set using the minimum and maximum rates. For example, James Smith runs a drugs and alcohol rehabilitation clinic. The organization pays James Smith at a fixed rate for the period that clients are placed in his clinic. Similarly, a maximum and minimum rate can be established for services within which the organization is willing to pay.

### ***Configuring Exceptional Payment Processing Options***

The following exceptional payment options can be configured for service offerings at the time of registration, or at any time after the service offering has been registered: minimum amount payment, maximum amount payment and fixed amount payment. These payment option settings are used to specify how to handle exceptional payments. For each of these options, the administrator can configure whether the payment is paid at a fixed amount, not paid, or paid as billed.

For example, if the fixed amount for a service offering payment is \$100, but the actual amount billed by the provider for the service offering is \$110, the payment option chosen for the service offering will dictate what the provider receives: The provider will be paid \$100 if 'pay fixed amount' is set, \$110 for 'pay as billed', or \$0 for 'don't pay'.

Exceptional payment options can also be set in the resource manager application for individual providers. This allows for a flexible approach to paying providers. Exceptional payment options set for individual providers takes precedence over those set at the service offering level in the administration application.

### ***Basing Payments on Placements or Attendances***

Payments for provider service offerings can be based on the number of client placements made with the provider or based on client attendance. Once service offering payments are configured to be placement-based or attendance based, it is not possible to change the setting.

In order for payments to be placement based, the unit of measure for the service offering must be place. A placement payment frequency can be set which denotes the frequency with which providers are paid for client placements. In contrast to using service invoices, payments based on placement data do not require service authorizations, or a request for payment from the provider. For example, an alcohol and drugs rehabilitation service is configured to be paid based on placements, with a payment frequency of weekly. A provider, Margaret Williams, offers the service and has three clients placed with her for the full week. Payment is automatically made at the end of the week, based on the placement data, for 21 units (3 clients for 7 days each).

Service offerings payments that are attendance based are calculated using rosters. An attendance payment frequency can be set which specifies the frequency with which providers are paid for attendance tracking enabled service offerings. For example, a case worker has authorized James Smith to receive child care from 9 AM to 5 PM on every Monday for the month of January from Dorothy Williams who is a registered child care provider. The service authorization is then recorded in the system. When payment is due for the service for the month of January, the system compares the planned hours from the service authorization and the actual hours attended which have been submitted through attendance tracking and makes the payment accordingly.

When a service offering is configured to be placement or attendance based, any providers who are offering this service will be paid based on how the service offering is configured. The resource

manager can, however, further customize the service offering to suit individual providers by setting things like provider specific service rates and payment options.

## Registering Training Service Offerings

Recording a training as a service offering is similar to the way in which a standard service offering is registered. If the training that is to be added as a training service offering is already recorded on the system, the administrator can create a training service offering using the existing training. If the training is not already recorded in the system, once the training name and type have been entered, it will automatically be saved as a training. Training credits can be added to the training at the time of registration or at a later point. Once the training service offering is recorded in the system, it can be added to a provider member's profile in the resource manager application.

### ***Configuring Training Requirements for Service Offerings***

Training requirements ensure that provider members are qualified to provide services to the organization's clients. Once a training is recorded on the system, it can be added as a training requirement for a service offering. If the training requirement has been added for the service offering, the resource manager can select the training from the list of training requirements and apply it to the provider member. Training requirements can be set as required or recommended to be completed by provider members.

For example, Linda Smith is a child minder who is employed by the provider, Better Child Care Ltd. In order to provide child care to the organization's clients, Linda is required to have completed an emergency first aid training course. If Linda has not completed the course, and the course is managed by the organization, she can be enrolled in it. If it is a non-managed training program, the details of the course can also be recorded. Once it is established that Linda has completed the course, she can provide child care on behalf of the organization.

## Configuring Service Offering Financials

The financial settings for service offerings are configured as part of the registration process. There are a number of system configuration options available which provide flexibility in how providers are paid for their services.

### ***Indicating if Manual Approval Is Required for Service Invoice Line Items***

The `curam.sili.manual.approval` property is used to define whether a service invoice line item requires manual approval or whether it is automatically approved upon submission. The default value is `TRUE`. When set to `true`, each service invoice line item will have to be approved by a supervisor before it can be used to process a provider payment. When set to `false`, the service invoice line item is automatically approved upon submission.

### ***Identifying the Product Provider Responsible for Paying Providers***

The `curam.financial.productproviderid` property is used to define the product provider for the product deliveries used for making payment to providers in CPM. Since it is the organization

that makes these payments, it might be suitable to have a product provider that denotes the organization assigned to this property.

### ***Raising an Event when an Invoice Based Payment is Withheld***

The `curam.financial.RaiseWithholdInvoicepaymentNotificationEvent` property is used to define whether an event is raised or not when an invoice based payment is withheld. This event can then be used to trigger further processing. The default value is `TRUE`.

### ***Configuring the Generate Payslips Batch Process***

The `curam.cpm.batch.generatepayslips.commitcount` property is used to define the commit count for the `GenerateCPMPayslips` batch program. Commit count processing defines the number of pay slip records, after which they are committed into the pay slip file. For example, if the value is set at 250, for every 250 payments that are created a batch program is run that generates the provider payslips.

The `curam.cpm.batch.generatepayslips.commitcountenabled` property is used to define whether the commit count processing in the `GenerateCPMPayslips` batch program will be used. If the property is disabled, payslips will be generated individually after all the payment processing is performed for a provider service.

## **Grouping Service Offerings**

Service groups are a set of similar services which are grouped together. For example, all foster care services (traditional foster care, special needs foster care) may be grouped together as foster care services.

## **Searching and Viewing Service Offerings**

The service offering search allows an administrator to search for and view service offerings based on the following search criteria:

- Service offering name - The search returns all service offerings whose name includes the text entered. For example, if searching for child, all service offerings with 'child' in their name are returned.
- Unit of measure - The search returns all service offerings matching the selected unit of measure. For example, if the place unit of measure is selected, all placement service offerings are returned.

## **Defining Approval Criteria for Service Offerings**

One of the important tasks of a CPM administrator is to configure approval criteria for service offerings. Approval criteria are certain conditions that providers and/or their provider members should have met before they are allowed to provide services to the organization's clients.

A number of different approval criteria can be selected for service offerings including background checks, accreditations, home studies, licenses, trainings, and other approval criteria required. When configured at the administrative level for a service offering, approval criteria automatically applies across all providers who offer the service.

Informational messages are displayed to warn the user in the event that any approval criteria have not been met. The user can choose whether or not to continue to approve the service offering. For example, if at the admin level, license and background checks are configured as approval criteria for a child care service offering, informational messages are displayed to the resource manager when attempting to approve a provider to deliver child care, where the required license or background checks are not in place.

When approving a service offering at the provider level, informational messages are displayed to warn the user that the necessary approval criteria has not been met. The user can choose whether or not to continue to approve the service offering. For example, if at the administrative level, license and background checks are configured as approval criteria for the child care service offering, informational messages are displayed to the user which detail the approval criteria that have not been met for provider licenses and background checks.

## **Preventing Incidental Authorization of Service Offerings**

The Prevent Authorization via Placement setting prevents a service authorization being created when a placement is made for a service. This is required because some services will be authorized via other means. This ensures that the authorization is not duplicated.

## **Keywords**

Keywords can be associated with service offerings which allows for easier searching of service when adding a service to an outcome plan. Note: This property is reserved for future use

## **Evidence**

Evidence can be added to a service on an outcome plan once an evidence type is associated with a service offering. An administrator can specify whether the evidence type is read only and also whether the evidence is stored against the person or the outcome plan.

## **Setting Up Attendance Tracking for Service Offerings**

Attendance tracking is used in CPM to track client participation, i.e., to confirm whether or not clients are attending the service offerings that are provided to them. It can be used to confirm that clients are successfully receiving the service offerings allocated to them. It can also be used to determine how much providers are paid based on client attendance.

### ***Roster Configuration Options***

Rosters are used to track client participation in service offerings. Rosters can be generated for each day specified in the roster generation frequency for every service offering that has attendance tracking configured. Multiple roster generation frequencies can be specified. However, only one roster generation frequency can be specified for a given time period and only one may be active at any one time. A roster generation frequency can also be specified for individual service offerings.

If manual approval is required for the roster, this can be set at the individual service offering level in the administration application. When manual approval is required, the resource manager will have to manually approve every roster upon submission. A roster grace period can also be specified. This is a period of time after the end of the roster period by which a provider must submit their roster for the service offering.

### **Setting up the Roster Grace Period**

The `curam.cpm.attendance.service.rostergraceperiod` property is used to set the roster grace period in days for a service offering. The grace period is the number of days after the roster submission date within which roster should be submitted to the agency for the service offering. If a roster is submitted after the specified grace period, a workflow event is raised. This workflow can be used to trigger any further customized processing that may be required by the organization in the event of a roster being submitted late.

For example: The roster grace period is set to 15 days for the Child Care service offering. If the roster submission date is 07/05/2008 then the roster due date for this roster containing clients receiving this service offering will be 15 days after this date, i.e., 07/20/2008.

The default value is 0.

### **Indicating if a Roster Submission Reminder is Required**

The `curam.cpm.attendance.service.submissionreminderrequired` property is used to determine whether or not a roster submission reminder for providers is required.

The default value is FALSE.

### **Setting up the Roster Submission Reminder Period**

The `curam.cpm.attendance.service.rostersubmissionreminderperiod` property determines the roster submission reminder period in days for a provider. This is the number of days prior to the roster submission due date, that the provider will be reminded about the roster submission due date. When a roster is due to be submitted, a notification is sent to the provider via their preferred communication method notifying them that the roster is due. The resource manager is also issued a notification.

The default value for this property is 0. This property is used only if the submission reminder required indicator is set to TRUE.

### **Indicating if Actual Roster Units Should Default to Expected Units**

The `curam.cpm.attendance.service.plannedunitsdefaulted` property determines whether the actual units of the roster should be defaulted to the expected units of a roster. If the property is set to TRUE, then the actual units for a period will be defaulted to the expected units. This means that the resource manager does not have to manually enter the units of the service that have been delivered by the provider.

If the units delivered are more than the authorized units, an exception task is created, the subject of which will in the following format: "The Actual units, <actual units>, must be less than or equal to the number of units authorized, <authorized units>."

The default value for this property is FALSE.

## Attendance Configuration Options

An attendance payment frequency and an effective date for the attendance payment frequency can be specified. Multiple attendance payment frequencies can be maintained for different service offerings. However, only one active attendance payment frequency can be specified at a time. When a new attendance payment frequency is specified, the previous frequency is superseded.

Further attendance tracking configuration options are available at the individual service offering level. Administrators can specify whether attendance tracking and/or daily attendance tracking is enabled for the service offering. There can only be one active attendance tracking configuration for a service offering at any one time.

## Recording Absence Reasons for Service Offerings

Absence reasons can be configured at the organization level and at the service offering level. The administrator can set an absence reason to be payable and/or deductible.

If an absence reason is configured as payable, this means that a payment will be made to the provider for a service in the event that that absence reason is recorded for that particular unit of the service offering. For example, if a client misses a service because of a family bereavement, and that absence reason is configured as payable, the provider will be paid for that unit of the service offering.

If an absence reason is configured as deductible, this means that in the event that a client misses a unit of the service for that absence reason, the service authorization will be updated accordingly by deducting a service authorization unit from the total units authorized. For example, if a client misses a service because of a unexplained absence, and that absence reason is configured as deductible, one unit will be deducted from the total units allowed on the service authorization.

## Indicating if Attendance Is Required for Payment

For each service offering, an attendance payment configuration can be specified. The Attendance Required for Payment setting is used to denote whether proof of client attendance (i.e., an attendance roster) is required in order for the provider to be paid for a service. Payments will not be made for an attendance based service paid based on invoices or a flat rate contract if a roster is not submitted and Attendance Required for Payment is set to Yes.

## 1.3 Service Evaluation Configuration

---

Evaluation of service deliveries involves answering questions related to various factors influencing the delivery such as staff attention, facility cleanliness etc. This serves two purposes - to determine the effectiveness of the delivery and to evaluate provider efficiency. Evaluating provider efficiency is the focus of service evaluation with respect to Provider Management (CPM).

Evaluation criteria are defined in the **Administration** application and the responses are captured from the service delivery. The responses are scored in order to determine the outcome of the evaluation.

## Configuring Evaluation Criteria

Evaluation criteria can be configured in the Administration application. Configuration involves specifying the name of the criterion, type of response for the criterion and the score for each response. Below are a few sample criteria.

### 1. Criterion - Was there adequate staff attention?

Response Type - Boolean.

Table 1: Scores

Response	Score
Yes	1
No	0

### 2. Criterion - Please rate the cleanliness of the facility.

Response Type - Rating 1 to 5.

Table 2: Scores

Response	Score
1	0
2	1
3	2
4	3
5	4

Scores are specified for the responses in order to determine the outcome of the service evaluation.

After setting up the evaluation criteria, these have to be related to the service offering (see Section 5.9 Configuring Evaluation Criteria for a Service Offering)

## 1.4 CPM Batch Processes

Various batch processes provide CPM functionality. Batch processes are configurable in the system administration application.

**Note:** Unless specified otherwise, batch processes may be run in any order.

## Batch Parameters and Processing Date

Quite a number of the batch jobs within the application take 'processingDate' as a parameter. When this is specified, it will be the date the batch job is run for, i.e., it will be the date returned from the getCurrentDate() API. When not specified, the batch job is run for the system date. This parameter is not typically listed as one of the parameters in the descriptions below.

## RosterSubmissionDue

This batch process is used to raise an event 'ROSTER.ROSTER\_NOT\_SUBMITTED\_AFTER\_DUE\_DATE' on the day after the roster submission due date, if the roster has not yet been submitted, i.e., the roster is in 'Normal' state. The event 'ROSTER.ROSTER\_NOT\_SUBMITTED\_AFTER\_DUE\_DATE' is used to send the notifications. The submission due date is calculated by adding the roster grace period to the last day of the roster period. The roster grace period is configured by the administrator for a service offering. Due to the nature of the batch job, it is expected that this batch run would be scheduled to coincide with the roster generation frequency. For example, if the roster generation frequency is monthly, and there is no roster grace period, then this batch job could be scheduled to be run the day after the last day of the roster period, i.e. the 1st of every month. If a number of different roster generation frequencies are in use, the batch could be run daily.

**Batch Process Class and Method** The class and method for this batch process is `RosterSubmissionDue.generateRosterNotification`.

## ProcessContractRenewalNotifications

This batch process is used to raise the workflow events 'ContractNotificationEvent.RENEWPROVIDERCONTRACT' and 'ContractNotificationEvent.RENEWPGCONTRACT' to send notifications to the user, a certain number of days prior to contract expiration. The number of days is configured by the system administrator using the 'curam.contracts.renewal.noofdaysevents' property. The batch process raises the workflow events when the difference between the contract end date and the current date is less than 'curam.contracts.renewal.noofdaysevents'. The workflow event 'ContractNotificationEvent.RENEWPROVIDERCONTRACT' is for the provider contract and the workflow event 'ContractNotificationEvent.RENEWPGCONTRACT' is for the provider group contract. This batch process run would be scheduled to run daily, or alternatively if an agency had a regular contract renewal date, the process could be scheduled to run a certain number of days prior to that date.

**Batch Process Class and Method** The class and method for this batch process is `ProcessContractRenewalNotifications.processContractRenewalNotifications`.

## ProcessLicenseRenewalNotifications

This batch process is used to raise a workflow event 'LicenseNotificationEvent.RENEWLICENSE', that sends a notification to the user a certain number of days prior to license expiration. The number of days is configured by the system administrator using the 'curam.license.renewal.noofdaysevents' property. The batch process raises the workflow event when the difference between the license end date and the current date is less than 'curam.license.renewal.noofdaysevents'. This batch process run would be scheduled to run daily, or alternatively if an agency had a regular license renewal date, the process could be scheduled to run a certain number of days prior to that date.

**Batch Process Class and Method** The class and method for this batch process is `ProcessLicenseRenewalNotifications.processLicenseRenewalNotifications`.

## ActivateCPMLiabilityCases

This batch process is used to activate the liability cases created as a part of CPM financial processing. The process retrieves all the liability cases created for the different financial product delivery cases in CPM (Invoice, Attendance, Contract and Placement), and sends them for processing so that they could be activated, if they are eligible for activation. This process then invokes the API 'ProductDeliveryActivationEligibility'. The API uses the method 'assessEligibilityForCase(CaseIDDetails)' to process the case for activation. This batch process would be scheduled to run daily and would be run before the financial batch processes 'GenerateInstructionLineItem' and 'GenerateInstruments' are run.

**Batch Process Class and Method** The class and method for this batch process is `ActivateCPMLiabilityCases.approveAndActivateCPMCases`.

## CreatePlacementEvidence

This batch process is used to create placement evidence for placements in respect of services that are configured to be paid based on placement data. The batch process first identifies the eligible placements, i.e., those placements for which evidence needs to be created. If the placement end date is not specified, the batch process uses the placement payment frequency and calculates the next occurrence till the next payment date. If the placement end date is specified and if it happens to be before the next payment date, then the batch process considers the placement end date and calculates the next occurrence till the placement end date.

For example, if there is a placement from 1st March to 15th April and the payment date is on every 29th, then the system should create placement evidences from 1st March to 29th March and 30th March to 15th April. Where evidence doesn't already exist, then it is created and added to the CPM Placement product delivery case, which triggers reassessment of the case. The batch processes 'CreatePlacementEvidence' and 'ReassessOutstandingCases' have to be scheduled to run before running 'GenerateInstructionLineItem' and 'GenerateInstrument' batch processes for placement based payments. The financials batch jobs 'GenerateInstructionLineItem' and 'GenerateInstruments' can then be used to generate payments to providers.

**Batch Process Class and Method** The class and method for this batch process is `CreatePlacementEvidence.createPlacementEvidence`.

## ExpireReservation

This batch process is used to expire active reservations when their date of expiry is reached. The reservation expiry date is calculated by adding the reservation grace period to the start date of the reservation. If the reservation expiry date for an active reservation is earlier than the processing date, then the active reservation is expired. The reservation grace period is set for the provider

and is configured by the user. If the reservation grace period is not set at the provider level, then the system considers the grace period set at the agency level. The grace period at the agency level is configured by the system administrator using the 'curam.cpm.reservation.agencygraceperiod' property. After the active reservation is expired, a reservation expiration notification is sent to the case owner of the case, stating that the reservation has expired. It is expected that this batch process would be scheduled to run daily.

The batch process takes no parameters.

**Batch Process Class and Method** The class and method for this batch process is `ExpireReservation.expireReservation`.

## GenerateCPMPayslips

This batch process is used to generate pay slips for all the un-issued pay slips (Invoice based, Attendance based, Placement and Flat-Rate Contract) from the payslip table. The batch process uses the property 'curam.cpm.batch.generatepayslips.commitcountenabled' to determine whether the batch process must use the commit count for processing and the property 'curam.cpm.batch.generatepayslips.commitcount' to determine the number of records to be processed by the batch before the results are committed to the database. Both these properties are configured by the system administrator. This batch process would be scheduled to run after the financial batch processes 'GenerateInstructionLineItem' and 'GenerateInstruments' are run.

**Batch Process Class and Method** The class and method for this batch process is `GeneratePayslips.generateNewPayslips`.

## GenerateRosters

This batch process is used to generate attendance rosters for providers for which service authorizations exist, i.e. they have clients expected to receive services from them in the upcoming period. The process will also generate a roster line item for every client authorized in the service authorization line item for the period of the roster. The batch process would be scheduled to run based on the roster generation frequency, configured for the provider. For example, if the roster generation frequency is monthly, then the batch would be scheduled to run on the 1st of every month. For more information on entering attendance and processing of rosters, please refer to the Provider Management Guide.

**Batch Process Class and Method** The class and method for this batch process is `GenerateRosters.generateRosters`.

## RosterSubmissionReminder

This batch process is used to generate reminder notifications which have not been generated previously for roster submission, by either using the XSL or the XML templates for the roster event `RosterNotificationEvent.ROSTER_SUBMISSION_REMINDER`. When the 'Roster Submission Reminder Required' indicator is set to true, the system sends the notifications to

the relevant providers, a certain number of days (configured by the user) prior to the roster submission due date, only if the reminder has not already been sent.

The 'Roster Submission Reminder Required' indicator is configured by the system administrator using 'curam.cpm.attendance.service.submissionreminderrequired' property. When this indicator is set to true, then the system considers 'curam.cpm.attendance.service.rostersubmissionreminderperiod' property to get the number of days prior to which the reminders have to be sent. This property is also configured by the system administrator. For example, if this batch process is run on 18th and the property 'curam.cpm.attendance.service.rostersubmissionreminderperiod' is set to 2, then the notifications are sent for all the rosters that have submission due date as 20th. It is expected that this batch process would be scheduled to run daily.

**Batch Process Class and Method** The class and method for this batch process is `RosterSubmissionReminder.createNotification`.

## ActivateProviderPaymentCorrectionCases

This batch process is used to process and activate the payment correction cases which are used to manage over and underpayments when CPM financial product deliveries (Invoice, Attendance, Contract and Placement) are configured not to use rolled up reassessment. The process retrieves all the payment correction cases created for the product delivery cases in each of the four financial product deliveries. The process then processes and activates the payment correction cases using the API 'paymentCorrection'. This batch process would be scheduled to run daily before the financial batch processes 'GenerateInstructionLineItem' and 'GenerateInstruments' are run.

**Batch Process Class and Method** The class and method for this batch process is `ActivateProviderPaymentCorrectionCases.processPaymentCorrectionCases`.

## ImportTaxonomyData

This batch process is used to import a taxonomy xml file into the system. The process is run each time after a new or the updated taxonomy xml file is available. When importing, the system checks the version numbers of any previously imported taxonomy versions against the version number of the taxonomy file that is being imported. The version number of the file is derived from the release date and the country code. If a version of the taxonomy which is imported already exists in the system with a different language (i.e. the two files are translations of the same taxonomy version), then the translations from the file being imported are added to the existing taxonomy. When a later version of a taxonomy is imported, the system compares the existing and new versions, and identifies any changes. This process is described in more detail in the Provider Management Guide.

**Batch Process Class and Method** The class and method for this batch process is `ImportTaxonomyData.importTaxonomy`.

 **(deprecated) PublishTaxonomyData**

This batch process uses the method 'publishTaxonomyData(PublishTaxonomyVersionKey)' within the class 'PublishTaxonomyData', to publish the taxonomy version. The method 'publishTaxonomyData(PublishTaxonomyVersionKey)' is used to publish the taxonomy version which is in 'Submitted For Publishing' status. The parameter 'PublishTaxonomyVersionKey' contains the batch processing date and the taxonomy version ID. The process creates related concepts, external system classification terms and taxonomy terms, and an association of taxonomy terms with the related concepts, external system classifications, related terms and term old codes. The batch process is run each time a new taxonomy file is imported and after making any required customizations. For more information, please refer to the Provider Management Guide.

**Batch Process Class and Method** The class and method for this batch process is `PublishTaxonomyData.publishTaxonomyData`.

**TaxonomyDatabaseSearchExtractor**

This batch process is used to extract the taxonomy application data from the application database and populate it to the staging database, from where it can be accessed much faster by the application. This batch job is scheduled to run each time after the taxonomy is published.

**Batch Process Class and Method** The class and method for this batch process is `TaxonomyDatabaseSearchExtractor.extractTaxonomyData`.

**1.5 Summary of CPM Application Properties**

Application properties are variables that are used to alter the functionality provided by the system at run-time, thereby providing a mechanism of changing functionality dynamically without the necessity of going through a full development cycle. The Provider Management application properties are configurable in the system administration application.

**External User Security Settings**

Table 3: Summary of External User Security Properties

Application Property	Description
<code>curam.custom.externalaccess.implementation</code>	This property specifies the CPM specific implementation file for external access security. The default value is <code>curam.cpm.eua.facade.impl.CPMExternalAccessSecurity</code> , which is the default CPM implementation.

## Application – CPM Attendance Settings

Table 4: Summary of CPM Attendance Properties

Application Property	Description
<code>curam.cpm.attendance.service.rostergraceperiod</code>	This property sets the roster grace period in days for a service offering. For example: The roster grace period is set to 15 days for the Child Care service offering. If the roster submission date is 07/05/2008 then the roster due date for this roster containing clients receiving this service offering will be 15 days after this date, i.e., 07/20/2008. The default value is 0.
<code>curam.cpm.attendance.service.plannedunitsdefaulted</code>	This property determines whether the actual units of the roster should be defaulted to the expected units of a roster. The default value is FALSE.
<code>curam.cpm.attendance.service.rostersubmissionreminderperiod</code>	This property determines the roster submission reminder period in days for a provider. This is the number of days prior to the roster submission due date, that the provider will be reminded about the roster submission due date. This property is used only if the submission reminder required indicator is set to TRUE. The default value is 0.
<code>curam.cpm.attendance.service.submissionreminderrequired</code>	This property determines whether or not a roster submission period is required. The default value is FALSE.
<code>curam.cpm.attendance.service.reporting.configured</code>	This property determines whether the reporting method on the service offering attendance configuration is mandatory or optional. The default value is TRUE.

## Application – CPM Financials Settings

Table 5: Summary of CPM Financial Properties

Application Property	Description
<code>curam.financial.financialhook.registrars</code>	This property lists the financial hook registrars. This is overridden from Cúram Platform to add the financial hook used by CPM. In a customer deployment where there could be more hook implementations, the hook from CPM should always be present in this list in order for CPM payments to work correctly. The CPM entry is <code>curam.cpm.sl.impl.CPMFinancialComponentProcessingRegistrarFactory</code> .
<code>curam.cpm.batch.generatepayslips.commitcountenabled</code>	This property defines whether the commit count processing in the GenerateCPMPayslips batch program will be used. The default value is YES.
<code>curam.cpm.batch.generatepayslips.commitcount</code>	This property defines the commit count for the GenerateCPMPayslips batch program. This means that for every 100 payslips that are created a batch program is run that generates the payslips for providers. The default value is 100.
<code>curam.sili.manual.approval</code>	This property defines the manual or automatic approval of a service invoice line item. The default value is TRUE.
<code>curam.financial.productproviderid</code>	This property defines the product provider for the product deliveries used for making payment to providers in CPM. Since it is the organization that makes these payments, it might be suitable to have a product provider that denotes the organization assigned to this property.
<code>curam.financial.RaiseWithholdInvoicepaymentNotificationEvent</code>	This property defines whether an event is raised or not when an invoice based payment is withheld. This event can then be used to trigger further processing. The default value is TRUE.
<code>curam.financial.RaisewithholdcontractpaymentNotificationEvent</code>	This property defines whether an event is raised or not when a contract based payment is withheld. This event can then be used to trigger further processing. The default value is TRUE.

Application Property	Description
<i>curam.financial.providerlocationid</i>	This property defines the provider location ID. The default value is 2701.
<i>curam.financial.cpm.UnlimitedAmountRequired</i>	This property defines if an agency wishes to have unlimited payable amount. If it is set to TRUE, there is no restriction on how much amount can be paid. If it is set to FALSE, the payable amount will be suspended if it is more than a specified limit. The default value is TRUE.
<i>curam.financial.FutureDatedSLIEnabled</i>	This property defines if an agency wishes to allow the submission of invoices for future dated services. If it is set to TRUE, users can submit invoices with a future service date. If it is set to FALSE, users will be prevented from submitting invoices with a future service date. The default value is TRUE.
<i>curam.financial.FutureDatedRostersEnabled</i>	This property defines if an agency wishes to allow the submission of rosters and roster line items containing future dated services. If it is set to TRUE, users can submit rosters and roster line items with a future service date. They can also enter absence or attendance details for future dates. If it is set to FALSE, users will be prevented from submitting rosters and roster line items with a future service date and from entering absence or attendance details for future dates. The default value is TRUE.
<i>curam.financial.CPMSysAdmin</i>	This property defines the owner and supervisor for any CPM product delivery cases. CPM System Administrator will act as the default case owner/supervisor. Once a CPM product delivery case is approved and activated internally, CPM System Administrator is set to be the case owner as well as the supervisor for the cases. This user can be changed by entering the exact username (primary key of the USERS table) in this field. A user should always be specified.
<i>curam.financial.affectedserviceinvoices</i>	This property defines whether service invoice financial processing will store a list of affected service invoices when corrections are made, such as to a service invoice line item or service rate, which will be used by the reassessment engine to restrict reassessment processing to a specific set of service invoices rather than processing all service invoices. If this application property is set to its default value of FALSE, reassessment processing will include all service invoice. It is recommended that this feature be used when the number of previous reassessments is large compared to the total number of invoices. When the number of previous reassessments is small compared to the total number of invoices, reassessment performance may not be optimum if this property is enabled.

## Application – CPM Facility Settings

Table 6: Summary of CPM Facility Properties

Application Property	Description
<i>curam.cpm.settings.maintainethnicitycharacteristics</i>	Indicates if Ethnicity characteristics can be maintained for a compartment within Provider Management. If it is set to false, users will not be able to maintain Ethnicity characteristics for a compartment. If set to true, users can maintain Ethnicity characteristics for a compartment. The default value is FALSE.
<i>curam.cpm.settings.maintainreligioncharacteristics</i>	Indicates if Religious characteristics can be maintained for a compartment within Provider Management. If it is set to false, users will not be able to maintain Religious characteristics for a compartment. If set to true, users can maintain Religious characteristics for a compartment. The default value is FALSE.

## Application - CPM Contracts Settings

Table 7: Summary of CPM Contract Properties

Application Property	Description
<code>curam.contracts.renewal.noofdaysevents</code>	This property sets the renewal period in days, for all the contracts within the organization. When this property is set, the system will automatically notify a user and provider a certain number of days prior to contract expiration. The default value is 7.

## Application CPM License Settings

Table 8: Summary of License Properties

Application Property	Description
<code>curam.cpm.license.renewal.noofdaysevents</code>	This property sets the renewal period in days, for all the licenses within the organization. When this property is set, the system will automatically notify a user and provider a certain number of days prior to license expiration. The default value is 7.

## Application - Sequential Continuous Reference Numbers

Table 9: Sequential Continuous Reference Numbers

Application Property	Description
<code>curam.provider.sequentialcontinuousreferencenumber</code>	This property holds the starting reference number for a provider. The default value is 10000000.
<code>curam.providergroup.sequentialcontinuousreferencenumber</code>	This property holds the starting reference number for a provider group. The default value is 20000000.
<code>curam.providerenquiry.sequentialcontinuousreferencenumber</code>	This property holds the starting reference number for a provider enquiry. The default value is 10000.
<code>curam.serviceinvoice.sequentialcontinuousreferencenumber</code>	This property holds the starting reference number for a service invoice. The default value is 10000.
<code>curam.serviceinvoicelineitem.sequentialcontinuousreferencenumber</code>	This property holds the starting reference number for a service invoice line item. The default value is 100000.
<code>curam.serviceinvoicerequestlineitem.sequentialcontinuousreferencenumber</code>	This property holds the starting reference number for a service invoice request line item. The default value is 100000.

## Application - Wait List Properties

Table 10: Wait List Application Properties

Application Property	Description
<code>curam.waitlist.statusreviewreminderperiod</code>	This property defines when an action task must be generated for wait list entry review. This would be n number of days before the expiry date. The value n is stored in the property. The default value is null.
<code>curam.waitlist.statusreviewintervalperiod</code>	This property defines the number of days between two reviews of a wait list entry. The default value is 0.

## Application - Miscellaneous Properties

Table 11: Miscellaneous Application Properties

Application Property	Description
<code>curam.cpm.reservation.agencygraceperiod</code>	This property defines an agency reservation grace period in days, from the planned start date, after which the reservation expires. The default value is 0.

## Infrastructure - Miscellaneous Properties

Table 12: Miscellaneous Infrastructure Properties

Application Property	Description
<code>curam.cpmprovider.cpmprovideremailserver</code>	This property defines an email server to be used for sending emails to provider/provider groups. The default value is <code>smtp.curamsoftware.com</code> .
<code>curam.miscapp.proximityCalculateDistanceIn</code>	This property defines the distance the calculated proximity between a providers address and a clients address is to be measured in. The value stored is a reference to an item in the unit of distance code table.

## Case Parameters

Table 13: Case Parameters

Application Property	Description
<code>curam.case.product.registrars</code>	This property lists the product registrars. This is overridden from Cúram Platform to add the Evidence Controller Hooks used by CPM. In a customer deployment where there might be more hook implementations, the hook from CPM must always be present in this list for CPM payments to work correctly. The CPM entry is <code>curam.financial.impl.FinancialRegistrar</code> .

## Application - CPM Dynamic SQL Enabled

Table 14: CPM Dynamic SQL Enabled

Application Property	Description
<code>curam.cpm.dynamicsql</code>	This property indicates whether or not the dynamic SQL functionality is enabled. The default value is YES.

# Notices

---

Permissions for the use of these publications are granted subject to the following terms and conditions.

## **Applicability**

These terms and conditions are in addition to any terms of use for the Merative website.

## **Personal use**

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of Merative

## **Commercial use**

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of Merative.

## **Rights**

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

Merative reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by Merative, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

MERATIVE MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

Merative or its licensors may have patents or pending patent applications covering subject matter described in this document. The furnishing of this documentation does not grant you any license to these patents.

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

Any references in this information to non-Merative 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 Merative product and use of those websites is at your own risk.

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.

The licensed program described in this document and all licensed material available for it are provided by Merative under terms of the Merative Client Agreement.

#### **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 Merative, 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. Merative, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. Merative shall not be liable for any damages arising out of your use of the sample programs.

## ***Privacy policy***

---

The Merative privacy policy is available at <https://www.merative.com/privacy>.

## ***Trademarks***

---

Merative™ and the Merative™ logo are trademarks of Merative US L.P. in the United States and other countries.

IBM®, the IBM® logo, and ibm.com® are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide.

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.

Oracle and Java are 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™, 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 company, product, and service names may be trademarks or service marks of others.