IBM Cúram Social Program Management

Cúram Provider Management Developers Guide

Version 6.0.5

IBM Cúram Social Program Management

Cúram Provider Management Developers Guide

Version 6.0.5

Note
Before using this information and the product it supports, read the information in "Notices" on page 67

Revised: May 2013

This edition applies to IBM Cúram Social Program Management v6.0 5 and to all subsequent releases unless otherwise indicated in new editions.

Licensed Materials - Property of IBM.

Contents

Figures

Tables

Chapter 1. Introduction

1.1 Purpose

The purpose of this guide is to describe the options for customizing the Cúram Provider ManagementCúram Provider Management (CPM) component. Its scope includes the customization of Strategy Patterns, Events, Workflows, Products, and Rule Sets. Customization can be distinguished from configuration in that customization allows developers to modify, extend, or replace source code to suit agency's requirements. Configuration allows administrators to manage the information that is displayed on application pages or to alter the behavior of the application in certain predefined ways.

This guide describes the customization or extension points provided with the CPM component. For further information on how to uses these customization points, readers should consult the `Persistence Cookbook` where customization techniques are described in detail.

This guide does not cover generic extension points such as persistence events.

1.2 Audience

This guide is intended for developers responsible for customizing CPM.

1.3 Prerequisite Reading

- `Persistence Cookbook`
- `Cúram Workflow Reference Guide`

1.4 Further Reading

For more information about the classes, interfaces, business event interfaces and the standard infrastructural persistence events included in the CPM component, please consult the CPM JavaDoc.

1.5 Chapters in this Guide

Using Strategy Patterns to Customize CPM
This chapter provides a brief description of the types of customization points that are available in CPM and how agencies can use them in a supported manner.

Using Events to Add Custom Processing to CPM
This chapter describes the events that can be used by agencies to add functionality before and/or after a piece of functionality is executed.

CPM Workflow Process Definitions
This chapter lists and details CPM Workflows that can be customized by the agency

CPM Products and Rule Sets
This chapter describes CPM products and Rule Sets.

CPM Financials
This chapter describes CPM Financials.

CPM Taxonomy
This chapter describes CPM Taxonomy.

Compliancy
This chapter describes compliancy information that should be kept in mind before any customization.

Appendix

This chapter provides appendixes.

Chapter 2. Using Strategy Patterns to Customize CPM

2.1 Introduction

CPM provides a number of service layer interfaces that are specifically designed for customization. A new custom implementation can be provided for any of the interfaces listed below. It is worth noting that default implementations are provided for these interfaces. Please read the Persistence Cookbook on how to provide an implementation for a service layer interface. The default implementations of these interfaces can be replaced with a new custom implementation by creating a new Guice module class and adding a corresponding entry in the MODULE table. Chapter 5 - Creating a Google Guice Module - in the Persistence Cookbook explains this in detail.

2.2 Provider Implementations

Table 1. Provider Implementations.

This table describes customizable implementations.

Functionality	Interface	Description
Provider Enrollment Date	curam.provider.impl.Provider	This interface allows agencies to enroll a Provider or Provider Group in CPM with an enrollment date which is in the past. This allows the agency to use the original date of enrollment while enrolling providers. The default implementation is to use the current date.
Provider Reference Number	curam.provider.impl.ProviderReferenceNumberStrategy	This interface allows agencies to generate Provider Reference Numbers according to their preferred format.
Provider Group Reference Number	curam.provider.impl.ProviderGroupReferenceNumberStrategy	This interface allows agencies to generate Provider Group Reference Numbers according to their preferred format.
Provider Enquiry Reference Number	curam.provider.impl.ProviderEnquiryReferenceNumberStrategy	This interface allows agencies to generate Provider Enquiry Reference Numbers according to their preferred format.
License Reference Number Generation	curam.provider.impl.LicenseNumberGenerator	This interface allows agencies to generate License Reference Numbers according to their preferred format.
Home Study Recommendation Approval	curam.provider.impl.ProviderSecurity	This interface allows agencies to designate a specific user or a group of users (an organization unit, users in a particular position or with a particular job, etc.) who can approve or reject a home study recommendation.
Provider Offering Approval Criteria	curam.providerservice.impl.ProviderOfferingApprovalCriteria	This interface allows agencies to specify criteria which need to be met in order to approve a service offered by a provider.

Table 1. Provider Implementations (continued).

This table describes customizable implementations.

Functionality	Interface	Description
Service Offering Validation	curam.serviceoffering.impl.ServiceOfferingValidation	A ServiceOfferingValidation class is used for managing the validations for a service. The default implementation of this interface is provided by ServiceOfferingValidationImpl. A new implementation of this interface is required to change the mechanism used to manage the validations for a service. This interface allows agencies to backdate the start date of a service offering. This may be useful when an agency is unable to add all provider services at the time of enrollment. This interface allows the agency to add a service at a later stage, and indicate that it has always been offered by the provider. The default date can be overridden on case-by-case basis.
External User Password	curam.externaluseraccess.impl.ExternalUserPasswordStrategy	This interface allows agencies to implement a particular strategy for allocating passwords, at the point at which they generate the initial password for a new external user account, or generate a replacement password for a user who has forgotten password and needs to re-establish credentials with the agency.
Provider Member Offering Training Criteria	curam.provider.impl.ProviderMemberOfferingTrainingCriteria	This interface allows agencies to change the default functionality when a provider offering with training requirements is approved. For example, the agency may wish to prevent the approval of a provider offering, if the training requirements for the service are neither 'Complete' nor 'Waived', rather than sending a notification.

2.3 Placement and Contract Implementations

Table 2. Placement and Contract Implementations.

This table describes customizable Placement and Contract implementations.

Functionality	Interface	Description
Placement Payment	curam.place.impl.PlacementPaymentStrategy	A PlacementPaymentStrategy class is used for determining if a placement is paid on the basis of an invoice or placement. The default implementation of this interface is provided by PlacementPaymentStrategyImpl. A new implementation of this interface is required to change the mechanism used to determine if a placement is paid on the basis of an invoice or placement. For example, an agency may indicate that all placement services should be paid through the receipt of invoices, or it may indicate that only services in a specific service group or services provided by specific providers can be paid through the receipt of invoices.
Flat Rate Contract Cover Pattern	curam.contracts.impl.FlatRateContractCoverPatternStrategy	A FlatRateContractCoverPatternStrategy class is used for determining the cover period pattern for a provider flat rate contract payment. The default implementation of this interface is provided by FlatRateContractCoverPatternStrategyImpl. A new implementation of this interface is required to change the mechanism used to determine the cover period pattern for a provider flat rate contract payment. A cover period pattern specifies how payments or bills are issued, e.g., in advance, in arrears, once-off, etc.
Contract Reference Number Generation	curam.contracts.impl.ReferenceNumberGenerator	A ReferenceNumberGenerator class is used for generating a reference number for a contract. The default implementation of this interface is provided by UniqueNumberGeneratorImpl. A new implementation of this interface is required to change the strategy to generate a reference number.

2.4 Training Implementations

Table 3. Training Implementations.

This table describes customizable Training implement ions

Functionality	Interface	Description
Approve License Based on Training	curam.provider.impl.LicenseApprovalCriteria	This interface allows agencies to change the default functionality for when a license with training requirements is approved. CPM supports notification to a user where training requirements for one or more services are neither 'Complete' nor 'Waived' for provider members. However, some Agencies may wish to prevent license approval if this validation is not satisfied. This interface is useful in such a scenario.
Approve Person Training	curam.training.impl.ApprovePersonTrainingProgramStrategy	This interface allows agencies to define their approval strategy for person training. The purpose of this interface is the same as that for the approval of provider member training, as described above.
Approve Provider Group Member Training	curam.training.impl.ApproveProviderGroupMemberTrainingProgramStrategy	This interface allows agencies to define their approval strategy for provider group member training. The purpose of this interface is the same as that for the approval of provider member training, as described above.
Approve Provider Member Training	curam.training.impl.ApproveProviderMemberTrainingProgramStrategy	This interface allows agencies to define their approval strategy for provider member training. CPM by default allows the resource manager or the resource manager supervisor to approve training.

The curam.training.impl.ApproveProviderMemberTrainingProgramStrategy interface can be used to facilitate functional scenarios such as the following:

- Agencies may choose to have another user or a group of users (an organization unit, users in a particular position or with a particular job) who can approve the training request.
- Agencies may wish to inhibit authorization of training based on some other additional or alternative approval criterion.
- CPM does not send any notification on approval of a training program. However, an agency may want to send a notification to the provider of the training, the provider the provider member works for, or the provider member themselves.

2.5 Service Invoice Implementations

Table 4. Service Invoice Implementations.

This table describes customizable Service Invoice implementations

Functionality	Interface	Description
Service Invoice Line Item	curam.financial.impl .ServiceInvoiceLineItemValidationStrategy	A ServiceInvoiceLineItemValidationStrategy class is used for validating the number of units of a service invoice line item. The default implementation of this interface is provided by ServiceInvoiceLineItem ValidationStrategyImpl. A new implementation of this interface is required to change the mechanism used to validate the number of units of a service invoice line item.
Service Invoice Payment	curam.financial.impl .ServiceInvoicePaymentStrategy	A ServiceInvoicePaymentStrategy class is used for managing service invoice payment strategy. The default implementation of this interface is provided by ServiceInvoicePaymentStrategyImpl. A new implementation of this interface is required to change the mechanism used to manage service invoice payment strategy. This may be useful where an agency wishes to re-direct these payments to an individual or a group other than the provider. For example, if the provider is specified as the payee on a service invoice line item but is an active member of a provider group, the agency may re-direct payments to the provider group instead.
Service Invoice Line Item Validation	curam.financial.impl .ServiceInvoiceLineItemValidator	A ServiceInvoiceLineItemValidator class is used for validating the service invoice line item. The default implementation of this interface is provided by ServiceInvoiceLineItemValidatorImpl. A new implementation of this interface is required to change the mechanism used to validate the service invoice line item. For example, some agencies may not want to allow a service invoice line item to be added to a service invoice if the status of the service invoice is 'In Progress'. This interface will allow them to implement this validation.

Table 4. Service Invoice Implementations (continued).

This table describes customizable Service Invoice implementations

Functionality	Interface	Description
Determine Service Invoice Line Item Payment Amount	curam.financial.impl .DeterminePaymentAmount	This business interface provides a mechanism for determining the amount to be paid for Service Invoice Line Item. If an Agency has a specific way in which they will want to calculate the payment amount, the customized implementation can be provided for this interface.

2.6 Custom Rates and Reassessment

Table 5. Custom Rates and Reassessment Implementations.

This table describes custom rates and reassessment implementations

Functionality	Interface	Description
Applicable Rate Listener	curam.financial.impl.ApplicableRateListener	This business interface is used for re-assessment of payments for a given period for a service authorization line item/placement/service invoice line item/provider roster line item. There are two APIs present in ApplicableRateListener having same name as "reAssess" but with different input types. 1. reassess API having inputs as Service Authorization Line Item and date range is used when no detailed product information is available and only Service Authorization Line Item is known. It searches to retrieve matching Service Invoice Line items, Placements and Provider Roster Line Items for the given Service Authorization Line Item and then it calls the suitable API present in ApplicableRateProcessor API to process the change of rate for any given input(placement /SILI/PRLI) and reassess the payment. 2. reassess API having inputs as Delivery Evidence Information of the product and date range is used when more product level information is available and the type of service is known. Depending on the product type it calls the suitable API present in ApplicableRateProcessor API to process the change of rate for any given input(placement /SILI/PRLI) and reassess the payment.

Table 5. Custom Rates and Reassessment Implementations (continued).

This table describes custom rates and reassessment implementations

Functionality	Interface	Description
Applicable Rate Processor	curam.financial.impl.ApplicableRateProcessor	An ApplicableRateProcessor class is used for reassessment of payments triggered by the change in rates. The default implementation of this interface is provided by ApplicableRateProcessorImpl. A new implementation of this interface is required to change the mechanism used to calculate the reassessed payment amount, due to the change in rates for the reassessment period. This interface allows agencies to process the change of rate for any given input (placement/SILI/PRLI) and reassess the payment. There are three APIs present in ApplicableRateProcessor named processRateChangeForPlacement, processRateChangeForPRLI and processRateChangeForSILI respectively. All these APIs are having inputs as type of service (Placement, SILI, PRLI) and the reassessment period. It processes the change of rate for any given input(placement /SILI/PRLI) and reassess the payment.
Service Delivery Rate Determination	curam.financial.impl.RateDetermination	A RateDetermination class is used for retrieving the rates for the given period and product delivery. The default implementation of this interface is provided by RateDeterminationImpl. A new implementation of this interface is required to change the strategy to determine the rates for a given delivery type (placement, invoice, or attendance) for a given period of time. For example, the applicable rates for a service can be determined using a custom rate calculation logic which may reference variables that do not reside within CPM, such as the number of children in a family.

2.7 Roster Implementations

Table 6. Roster Implementations.

This table describes customizable Roster implementations

Functionality	Interface	Description
Generate Rosters	curam.attendance.impl.DetermineRosterSubmissionDueDate	DetermineRosterSubmissionDueDate class is used for determination of submission due date for a roster. The default implementation of this interface is provided by DetermineRosterSubmissionDueDateImpl. A new implementation of this interface is required to change the way the grace period is used to determine the submission due date. For instance, an agency may wish to consider only the business days to calculate a submission due date.
Match Provider Roster Line Item	curam.attendance.impl.MatchProviderRosterLineItem	A MatchProviderRosterLineItem class is used for performing validations during matching a provider roster line item details with the existing details. The default implementation of this interface is provided by MatchProviderRosterLineItemImpl. A new implementation of this interface is required to change the mechanism used to match the details of a provider roster line item with the existing details. It is used for performing an agency's own program-specific validations during matching a provider roster line item.
Match Provider Roster Line Item	curam.attendance.impl.VoucherValidator	A VoucherValidator class is used for matching and validating the voucher details. The default implementation of this interface is provided by VoucherValidatorImpl. A new implementation of this interface is required to change the mechanism used to match and validate the voucher details of the provider roster line item. For example, the agency might have its own program-specific validations to match and validate the voucher details.

Table 6. Roster Implementations (continued).

This table describes customizable Roster implementations

Functionality	Interface	Description
Determine Attendance Based Payment Amount	curam.attendance.impl.AttendancePaymentDetermination Processing	AttendancePaymentDeterminationProcessing class is used for the determination of an attendance-based payment amount. The default implementation of this interface is provided by AttendancePaymentDeterminationProcessingImpl. A new implementation of this interface is required to change the mechanism used to calculate the attendance-based payment rate. For example, the provider service rate valid either on the end date of the roster line item or the end date of the matching service authorization line item could be used to determine the attendance-based payment amount.
Allocate Units	curam.attendance.impl.PRLIUnitsAllocationProcessing	A PRLIUnitsAllocationProcessing class is used for managing the allocation of units from roster line items to matching service authorization line items. The default implementation of this interface is provided by PRLIUnitsAllocationProcessingImpl. A new implementation of this interface is required to change the mechanism used to allocate units to the matching service authorization line items. For example, units could be allocated evenly to all service authorization line items rather than starting with the earliest one.

2.8 Taxonomy Search Implementations

Table 7. Taxonomy Search Implementations.

This table describes customizable Taxonomy Search implementations

Functionality	Interface	Description
Taxonomy term search based on different criteria	curam.taxonomy.sl.search.impl.TaxonomySearch	This business interface retrieves the taxonomy terms associated with provider service. This interface allows agencies to search the taxonomy terms based on different search criteria like term name, term code, indexed provider services etc. The default implementation uses the Generic Search Server (GSS) to fetch the search results.

2.9 Performance Measure Implementations

Table 8. Performance Measure Implementations.

This table describes customizable Performance Measure implementations

Functionality	Interface	Description
Retrieve Performance Measures	curam.performancemeasure.impl.RetrieveAllPerformanceMeasuresForProviderOffering	RetrieveAllPerformanceMeasuresForProviderOffering class is used for retrieving the performance measure's details for a provider service. The default implementation of this interface is provided by RetrieveAllPerformanceMeasuresForProviderOfferingImp A new implementation of this interface is required to change the mechanism used to determine the performance measure details of a provider service. Performance measure is the criteria by which the performance can be measured. For example, percentage of clients remaining in employment 1 year following the delivery of job search training.
Retrieve Performance Measures	curam.performancemeasure.impl.RetrieveAllPerformanceMeasuresForProvider	RetrieveAllPerformanceMeasuresForProvider class is used for retrieving the performance measure's details for a provider. The default implementation of this interface is provided by RetrieveAllPerformanceMeasuresForProviderImpl. A new implementation of this interface is required to change the mechanism used to determine the performance measure details of a provider. Performance measure is the criteria by which the performance of a provider can be measured. For example, percentage of clients remaining in employment 1 year following the delivery of job search training.
Retrieve Contract Performance Measure Value	curam.performancemeasure.impl.RetrieveAllContractPerformanceMeasures	RetrieveAllContractPerformanceMeasures class is used for retrieving the performance measure's details associated with a contract. The default implementation of this interface is provided by RetrieveAllContractPerformanceMeasuresImpl. A new implementation of this interface is required to change the mechanism used to determine the value for performance measure associated with the contract.

Chapter 3. Using Events to Add Custom Processing to CPM

3.1 Introduction

The sections below detail the events that are raised by CPM which allow developers to add custom functionality. Business events are raised at all extension points. These events can be used by agencies to add functionality before and/or after the action is executed.

3.2 Provider Customization Points

The following sections list the available customization points for Providers.

3.2.1 Provider Events

The following events are located in the curam.provider.impl.Provider interface.

Table 9. Provider Event Details.

This table describes Provider Events

Event Class	Description	Event is raised before and after
ProviderSuspendEvents	Raised when a Provider is suspended.	curam.provider.impl.Provider.suspend()
ProviderCloseEvents	Raised when a Provider is closed.	curam.provider.impl.Provider.close()
ProviderRejectEvents	Raised when a Provider seeking approval is rejected.	curam.provider.impl.Provider.reject()
ProviderApproveEvents	Raised when a Provider is approved.	curam.provider.impl.Provider.approve()
ProviderReopenEvents	Raised when a closed Provider is reopened.	curam.provider.impl.Provider.activate()
ProviderEnrollEvents	Raised when a Provider is enrolled.	curam.provider.impl.Provider.enroll()
ProviderGetAvailablePlacesInDateRangeEvents	Raised when available Places in the given date range are retrieved.	curam.provider.impl.Provider.getAvailablePlacesInDat
ProviderGetServiceOfferingsEvents	Raised when Service Offerings for a Provider are retrieved.	curam.provider.impl.Provider.getServiceOfferings()
ProviderGetCommonApprovedProviderServiceOfferingsEvents	Raised when Service Offerings for a Provider are retrieved.	curam.provider.impl.Provider.getCommonApprovedP

The following events are located in the curam.provider.impl.ProviderApprovalCheck interface.

Table 10. Provider Event Details.

This table describes Provider Events

Event Class	Description	Event is raised before and after
ProviderApprovalCheckCreateProviderApprovalCheckEvents	Raised when an approval check for the Provider is created.	curam.provider.impl.ProviderApprovalCheck.createPr
ProviderApprovalCheckModifyProviderApprovalCheckEvents	Raised when an approval check for the Provider is modified.	curam.provider.impl.ProviderApprovalCheck.modifyP
ProviderApprovalCheckCancelProviderApprovalCheckEvents	Raised when an approval check for the Provider is canceled.	curam.provider.impl.ProviderApprovalCheck.cancelPr

3.2.2 Provider Enquiry Events

The following events are located in the curam.provider.impl.ProviderEnquiry interface.

Table 11. Provider Enquiry Event Details.

This table describes Provider Enquiry Events

Event Class	Description	Event is raised before and after
ProviderEnquiryCloseEvents	Raised when a Provider Enquiry is closed.	curam.provider.impl.ProviderEnquiry.close()
ProviderEnquiryTransferEnquiryToProviderEvents	Raised when a Provider is enrolled from an enquiry.	curam.provider.impl.ProviderEnquiry.transferEnquiryToP
ProviderEnquirySetProviderEnquiryDetailsEvents	Raised when an enquiry is created.	curam.provider.impl.ProviderEnquiry.setProviderEnquiry
ProviderEnquirySetProviderEnquiryUpdateDetailsEvents	Raised when an enquiry is updated.	curam.provider.impl.ProviderEnquiry.setProviderEnquiry

3.2.3 License Events

The following Events are located in the curam.provider.impl.License interface.

Table 12. License Event Details.

This table describes License Events

Event Class	Description	Event is raised before and after
LicenseSuspendEvents	Raised when a License is suspended.	curam.provider.impl.License.suspend()
LicenseRejectEvents	Raised when a License is rejected.	curam.provider.impl.License.reject()
LicenseApproveEvents	Raised when a License approved.	curam.provider.impl.License.approve()

3.2.4 Home Study Events

The following Events are located in the curam.homestudy.impl.HomeStudy interface.

Table 13. Home Study Event Details.

This table describes Home Study Events

Event Class	Description	Event is raised before and after
HomeStudyApproveEvents	Raised when a Home Study recommendation for a provider is approved.	curam.homestudy.impl.HomeStudy.approve()
HomeStudySubmitEvents	Raised when a Home Study is submitted.	curam.homestudy.impl.HomeStudy.submit()
HomeStudyRejectEvents	Raised when a Home Study recommendation is rejected.	curam.homestudy.impl.HomeStudy.reject()

3.2.5 Compartment Events

The following Events are located in the curam.place.impl.Compartment interface.

Table 14. Compartment Event Details.

This table describes Compartment Events

Event Class	Description	Event is raised before and after
CompartmentCloseEvents	Raised when a Compartment is closed.	curam.place.impl.Compartment.close()

3.2.6 Place Events

The following events are located in the curam.place.impl.Place interface.

Table 15. Place Event Details.

This table describes Place Events

Event Class	Description	Event is raised before and after
PlaceCloseEvents	Raised when a Place is closed.	curam.place.impl.Place.activate()
PlaceMarkOutOfUseEvents	Raised when a Place is marked out of use.	curam.place.impl.Place.markOutOfUse()
PlaceOccupiedEvents	Raised when a Place is occupied.	curam.place.impl.Place.occupied()
PlaceMarkInUseEvents	Raised when a Place is marked in use.	curam.place.impl.Place.markInUse()
PlaceGetLocationForPlaceEvents	Raised when the location of a Place is retrieved.	uram.place.impl.Place.getLocationForPlace()

3.2.7 Request Events

The following events are located in the curam.externaluseraccess.impl.Request interface.

Table 16. Request Event Details.

This table describes Request Events

Event Class	Description	Event is raised before and after
RequestAcceptEvents	Raised when a Request created by an external provider is accepted.	curam.externaluseraccess.impl.Request.accept()
RequestSubmitEvents	Raised when Request created by an external provider is submitted.	curam.externaluseraccess.impl.Request.submit()
RequestRejectEvents	Raised when Request created by an external provider is rejected.	curam.externaluseraccess.impl.Request.reject()

3.2.8 Member Certification Events

The following events are located in the curam.provider.impl.MemberCertification interface.

Table 17. Member Certification Event Details.

This table describes Member Certification Events

Event Class	Description	Event is raised before and after
MemberCertificationModifyCertificationEvents	Raised when a provider member Certification is updated.	curam.provider.impl.MemberCertification.modifyCerti
MemberCertificationGetDerivedStstusEvents	Raised when the status of a provider member Certification is retrieved.	curam.provider.impl.MemberCertification.getDerivedS

3.2.9 Provider Deduction Events

The following events are located in the curam.provider.impl.ProviderDeduction interface.

Table 18. Provider Deduction Event Details.

This table describes Provider Deduction Events

Event Class	Description	Event is raised before and after
ProviderDeductionActivateProviderDeductionEvents	Raised when Deductions associated to a Provider are activated.	curam.provider.impl.ProviderDeduction.activateProvider
ProviderDeductionDeactivateProviderDeductionEvents	Raised when Deductions associated to a Provider are deactivated.	curam.provider.impl.ProviderDeduction.deactivateProvid
ProviderDeductionCreateDeductionForExistingCasesEvents	Raised when a Deduction is created for existing cases.	curam.provider.impl.ProviderDeduction.createDeduction
ProviderDeductionCreateVariableDeductionForModifiedPayeeDetTypeEvents	Raised when Variable Payee Deductions are created based on Payment Type.	curam.provider.impl.ProviderDeduction.createVariableDe
ProviderDeductionCancelVariableDeductionForModifiedPayeeDetTypeEvents	Raised when Variable Payee Deductions are Cancelled.	curam.provider.impl.ProviderDeduction.cancelVariableDe

3.2.10 Provider Offering Events

The following events are located in the curam.providerservice.impl.ProviderOffering interface.

Table 19. Provider Offering Event Details.

This table describes Provider Offering Events

Event Class	Description	Event is raised before and after
ProviderOfferingApproveEvents	Raised when a Provider Offering is approved.	curam.providerservice.impl.ProviderOffering.approve()
ProviderOfferingDenyEvents	Raised when a Provider Offering is denied.	curam.providerservice.impl.ProviderOffering.deny()
ProviderOfferingCheckApprovalCriteriaEvents	Raised when approval criteria are checked for a Provider Offering.	curam.providerservice.impl.ProviderOffering.checkAppro
ProviderOfferingGetContractsEvents	Raised when Contracts are retrieved for a Provider Offering.	curam.providerservice.impl.ProviderOffering.getContract

The following events are located in the curam.citizenactivity.impl.ProviderOfferingUtil interface.

Table 20. Provider Offering Event Details.

This table describes Provider Offering Events

Event Class	Description	Event is raised before and after
ProviderOfferingUtilGetByServiceOfferAndProviderEvents	Raised when a Provider Offering is retrieved based on the Service Offering and Provider.	curam.citizenactivity.impl.ProviderOfferingUtil.getByServ

3.2.11 Provider Offering Rate Events

The following events are located in the curam.providerservice.impl.ProviderOfferingRate interface.

Table 21. Provider Offering Rate Event Details.

This table describes Provider Offering Rate Events

Event Class	Description	Event is raised before and after
ProviderOfferingRateModifyForContractUsed	Raised when a Provider Offering Rate is modified.	curam.providerservice.impl.ProviderOfferingRate.mod

3.2.12 Performance Measure Events

The following events are located in the curam.performancemeasure.impl.RetrievePerformanceMeasureForProvider interface.

Table 22. Performance Measure Event Details.

This table describes Performance Measure Events

Event Class	Description	Event is raised before and after
RetrievePerformanceMeasureForProviderUsed	Raised when the Performance Measure on number of Incidents registered with the Provider is retrieved.	curam.performancemeasure.impl.RetrievePerformance
RetrievePerformanceMeasureForProviderUsed	Raised when the Performance Measure on number of investigations registered with the Provider is retrieved.	curam.performancemeasure.impl.RetrievePerformance
RetrievePerformanceMeasureForProviderUsed	Raised when the Performance Measure on number of Incidents requiring investigations registered with the Provider is retrieved.	curam.performancemeasure.impl.RetrievePerformance
RetrievePerformanceMeasureForProviderUsed	Raised when the actual value for a custom Performance Measure for a Provider is retrieved.	curam.performancemeasure.impl.RetrievePerformance

The following events are located in the curam.performancemeasure.impl.RetrievePerformanceMeasureForProviderOffering interface.

Table 23. Performance Measure Event Details.

This table describes Performance Measure Events

Event Class	Description	Event is raised before and after
RetrievePerformanceMeasureForProviderOfferingUsed	Raised when the Performance Measure on average cost per unit of service is retrieved.	curam.performancemeasure.impl.RetrievePerformance
RetrievePerformanceMeasureForProviderOfferingUsed	Raised when the Performance Measure on number of units per client is retrieved.	curam.performancemeasure.impl.RetrievePerformance
RetrievePerformanceMeasureForProviderOfferingUsed	Raised when the actual value for a custom Performance Measure for a Provider is retrieved.	curam.performancemeasure.impl.RetrievePerformance

3.3 Provider Group Customization Points

The following sections list the available customization points for Providers Groups.

3.3.1 Provider Group Events

The following events are located in the curam.provider.impl.ProviderGroup interface.

Table 24. Provider Group Event Details.

This table describes Provider Group Events

Event Class	Description	Event is raised before and after
ProviderGroupCloseEvents	Raised when a Provider Group is closed.	curam.provider.impl.ProviderGroup.close()
ProviderGroupEnrollEvents	Raised when a Provider Group is enrolled.	curam.provider.impl.ProviderGroup.enroll()
ProviderGroupReopenEvents	Raised when a closed Provider Group is reopened.	curam.provider.impl.ProviderGroup.reopen()
ProviderGroupGetCommonApprovedProviderServiceOfferingsEvents	Raised when approved Service Offerings of a Provider Group are retrieved.	curam.provider.impl.ProviderGroup.getCommonApprove

3.3.2 Provider Group Associate Events

The following events are located in the curam.provider.impl.ProviderGroupAssociate interface.

Table 25. Provider Group Associate Event Details.

This table describes Provider Group Associate Events

Event Class	Description	Event is raised before and after
ProviderGroupAssociateRemoveProviderEvents	Raised when a Provider is removed from a Provider Group.	curam.provider.impl.ProviderGroupAssociate.removePro

3.4 Service Offering Customization Points

The following sections list the available customization points for Service Offerings.

3.4.1 Service Offering Events

The following events are located in the curam.serviceoffering.impl.ServiceOffering interface.

Table 26. Service Offering Event Details.

This table describes Service Offering Events

Event Class	Description	Event is raised before and after
ServiceOfferingGetServiceRatesForPeriodEvents	Raised when Service rates are retrieved for a particular period.	curam.serviceoffering.impl.ServiceOffering.getServiceRat ()
ServiceOfferingModifyDescriptionTextTranslationEvents	Raised when the text translation details for the Service Offering description attribute is modified.	curam.serviceoffering.impl.ServiceOffering.modifyDescrip
ServiceOfferingModifyNameTextTranslationEvents	Raised when the text translation details for the Service Offering name attribute is modified.	curam.serviceoffering.impl.ServiceOffering.modifyNameT

Table 26. Service Offering Event Details (continued).

This table describes Service Offering Events

Event Class	Description	Event is raised before and after
ServiceOfferingAddNameTextTranslationEvents	Raised when the text translation is created for the Service Offering name attribute.	curam.serviceoffering.impl.ServiceOffering.addNameT
ServiceOfferingAddDescriptionTextTranslationEvents	Raised when the text translation is created for the Service Offering description attribute.	curam.serviceoffering.impl.ServiceOffering.addDescrip

3.4.2 Service Group Events

The following events are located in the curam.serviceoffering.impl.ServiceGroup interface.

Table 27. Service Group Event Details.

This table describes Service Group Events

Event Class	Description	Event is raised before and after
ServiceGroupAddServiceOfferingEvents	Raised when a Service Offering is added to a Service Group.	curam.serviceoffering.impl.ServiceGroup.addServiceO ()
ServiceGroupRemoveServiceOfferingEvents	Raised when a Service Offering is removed from a Service Group.	curam.serviceoffering.impl.ServiceGroup.removeServic ()
ServiceGroupGetServiceOfferingsEvents	Raised when Service Offerings from a Service Group are retrieved.	curam.serviceoffering.impl.ServiceGroup.getServiceOf ()
ServiceGroupRetrieveServiceGroupByReferenceEvents	Raised when the details of a Service Group for a specified reference is retrieved.	curam.serviceoffering.impl.ServiceGroup.retrieveServi

3.5 Service Authorization Customization Points

The following sections list the available customization points for Service Authorizations.

3.5.1 Service Authorization Events

The following events are located in the curam.serviceauthorization.impl.ServiceAuthorization interface.

Table 28. Service Authorization Event Details.

This table describes Service Authorization Customization Points

Event Class	Description	Event is raised before and after
ServiceAuthorizationFindLineItemByServiceEvents	Raised when Service Authorization Line Items for a particular Service are retrieved.	curam.serviceauthorization.impl.ServiceAuthorization
ServiceAuthorizationAddLineItemEvents	Raised when a line item is added to a Service Authorization.	curam.serviceauthorization.impl.ServiceAuthorization
ServiceAuthorizationInsertServiceAuthorizationEvents	Raised when a Service Authorization is created.	curam.serviceauthorization.impl.ServiceAuthorization
ServiceAuthorizationAddVoucherToServiceAuthorizationEvents	Raised when a voucher is associated to a Service Authorization.	curam.serviceauthorization.impl.ServiceAuthorization

Table 28. Service Authorization Event Details (continued).

This table describes Service Authorization Customization Points

Event Class	Description	Event is raised before and after
ServiceAuthorizationDeleteVoucherForSA	Raised when a voucher is disassociated with a Service Authorization.	curam.serviceauthorization.impl.ServiceAuthorization.del
ServiceAuthorizationGetDerivedStatusEvents	Raised when the status of a Service Authorization is retrieved.	curam.serviceauthorization.impl.ServiceAuthorization.get
ServiceAuthorizatnAddLineItemEvents	Raised when a specified line item is added to the Service Authorization.	curam.serviceauthorization.impl.ServiceAuthorization.ad
ServiceAuthorizationAddLineItemEvents	Raised when a specified line item is added to the Service Authorization.	curam.serviceauthorization.impl.ServiceAuthorization.ad
ServiceAuthorizationAddSALIToSAUsingFrequency	Raised when Service Authorization Line Items are generated and added to a Service Authorization based on the frequency pattern and date.	curam.serviceauthorization.impl.ServiceAuthorization.ad

3.5.2 Service Authorization Line Item Events

The following events are located in the curam.serviceauthorization.impl.ServiceAuthorizationLineItem interface.

Table 29. Service Authorization Line Item Event Details.

This table describes Service Authorization Line Item Events

Event Class	Description	Event is raised before and after
ServiceAuthorizationLineItemCloseEvents	Raised when a Service Authorization Line Item is closed.	curam.serviceauthorization.impl.ServiceAuthorizationLin
ServiceAuthorizationLineItemInsertServiceAuthorizationLineItemEvents	Raised when a Service Authorization Line Item is inserted.	curam.serviceauthorization.impl.ServiceAuthorizationLin
ServiceAuthorizationLineItemModifyServiceAuthorizationLineItemEvents	Raised when a Service Authorization Line Item is updated.	curam.serviceauthorization.impl.ServiceAuthorizationLin
ServiceAuthorizationLineItemCancelServiceAuthorizationLineItemEvents	Raised when a Service Authorization Line Item is cancelled.	curam.serviceauthorization.impl.ServiceAuthorizationLin
ServiceAuthorizationLineItemGetDerivedStatusEvents	Raised when the status of a Service Authorization is retrieved.	curam.serviceauthorization.impl.ServiceAuthorizationLin
ServiceAuthorizationLineItemGetRelatedLineItemEvents	Raised when Service Line Items related to a Service Authorization are retrieved.	curam.serviceauthorization.impl.ServiceAuthorizationLin

The following events are located in the curam.financial.impl.ProcessReassessmentForSALI interface.

Table 30. Service Authorization Line Item Event Details.

This table describes Service Authorization Line Item Events

Event Class	Description	Event is raised before and after
ProcessReassessmentForSALIReAssessOnCancellationEvents	Raised when the over payment for the Service Invoice Line Items or Provider Roster Line Items associated with the Service Authorization Line Item on cancellation of Service Authorization Line Item is processed.	curam.financial.impl.ProcessReassessmentForSALI.reA
ProcessReassessmentForSALIReAssessOnClosureEvents	Raised when the over payment on closing the Service Authorization Line Item is processed.	curam.financial.impl.ProcessReassessmentForSALI.reA
ProcessReassessmentForSALIReAssessOnCreationEvents	Raised when the reassessment on creation of new Service Authorization Line Item is triggered.	curam.financial.impl.ProcessReassessmentForSALI.reA
ProcessReassessmentForSALIReAssessOnModificationEvents	Raised when the reassessment on modification of new Service Authorization Line Item is triggered.	curam.financial.impl.ProcessReassessmentForSALI.reA

3.6 Placement Customization Points

The following sections list the available customization points for Placements.

3.6.1 Placement Events

The following events are located in the curam.place.impl.Placement interface.

Table 31. Placement Event Details.

This table describes Placement Events

Event Class	Description	Event is raised before and after
PlacementTransferClientEvents	Raised when a client is transferred to a new Place.	curam.place.impl.Placement.transferClient()
PlacementTransferClientToReservationEvents	Raised when a client is transferred to a new Place and a reservation is created for the new Place.	curam.place.impl.Placement.transferClientToReservati
PlacementGetOverlappingPlacementForClientEvents	Raised when overlapping Placement details for a client are retrieved.	curam.place.impl.Placement.getOverlappingPlacement

The following events are located in the curam.place.impl.FacilityInformation interface.

Table 32. Placement Event Details.

This table describes Placement Events

Event Class	Description	Event is raised before and after
FacilityInformationRetrieveFacilityInformationEvents	Raised when the the list of facility information for a Provider, service (or) provider type is retrieved.	curam.place.impl.FacilityInformation.retrieveFacilityIn

The following events are located in the curam.place.impl.PlaceSearch interface.

Table 33. Placement Event Details.

This table describes Placement Events

Event Class	Description	Event is raised before and after
PlaceSearchSearchAvailablePlacesEvents	Raised when an available Places in a Provider facility is searched.	curam.place.impl.PlaceSearch.searchAvailablePlaces()

3.6.2 Reservation Events

The following events are located in the curam.reservation.impl.Reservation interface.

Table 34. Reservation Event Details.

This table describes Reservation Events

Event Class	Description	Event is raised before and after
ReservationExpireEvents	Raised when a reservation is expired.	curam.reservation.impl.Reservation.expire()
ReservationCreateReservationEvents	Raised when a reservation is created.	curam.reservation.impl.Reservation.createReservation ()
ReservationConfirmPlacementEvents	Raised when a placement is created from a reservation.	curam.reservation.impl.Reservation.confirmPlacement ()
ReservationUpdateReservationEvents	Raised when a reservation is updated.	curam.reservation.impl.Reservation.updateReservation ()
ReservationCancelOverlappingActiveReservationEvents	Raised when overlapping active reservations are cancelled.	curam.reservation.impl.Reservation.cancelOverlappingAc ()
ReservationGetPlaceAvailableInDateRangeEvents	Raised when available placements in the given date range are retrieved.	curam.reservation.impl.Reservation.getPlaceAvailableInD ()

3.7 Contract Customization Points

The following sections list the available customization points for Contracts.

3.7.1 Contract Version Events

The following events are located in the curam.contracts.impl.ContractVersion interface.

Table 35. Contract Version Event Details.

This table describes Contract Version Events

Event Class	Description	Event is raised before and after
ContractVersionPrintContractEvents	Raised when a Contract Version is printed.	curam.contracts.impl.ContractVersion.printContract()
ContractVersionPreviewContractEvents	Raised when a Contract Version is previewed.	curam.contracts.impl.ContractVersion.previewContract()
ContractVersionValidateContractedProviderOfferingRatesEvents	Raised when Contracted Provider Offering rates are validated.	curam.contracts.impl.ContractVersion.validateContracted
ContractVersionValidateContractedProviderOfferingPlaceLimitsEvents	Raised when Contracted Provider Offering Place Limits are validated.	curam.contracts.impl.ContractVersion.validateContracted

The following events are located in the curam.contracts.impl.ContractVersionProviderOffering interface.

Table 36. Contract Version Event Details.

This table describes Contract Version Events

Event Class	Description	Event is raised before and after
ContractVerProvOfferCopyNonContractPORFromContractEvents	Raised when non contracted provider offering rates are copied to contract.	curam.contracts.impl.ContractVersionProviderOffering
ContractVerProvOfferCreateDefaultRateEvents	Raised when a default Provider Offering Rate is created for the Provider.	curam.contracts.impl.ContractVersionProviderOffering
ContractVerPOCheckForDuplicatePOOnLiveContractEvents	Raised when duplicate Provider Offering on live contract is checked.	curam.contracts.impl.ContractVersionProviderOffering
ContractVerPOCreateContractedPORFromNonContractedEvents	Raised when contracted Provider Offering Rate is created if the non contracted Provider Offering Rate does not exist.	curam.contracts.impl.ContractVersionProviderOffering

3.7.2 Flat Rate Contract Events

The following events are located in the curam.contracts.impl.FlatRateContract interface.

Table 37. Flat Rate Contract Event Details.

This table describes curam.contracts.impl.FlatRateContract

Event Class	Description	Event is raised before and after
FlatRateContractActivateEvents	This event is raised during activation of a flat rate contract.	curam.contracts.impl.FlatRateContract.activate()
curam.contracts.impl.FlatRateContract.FlatRateContractReEditEvents	Raised when a flat rate contract is edited.	curam.contracts.impl.FlatRateContract.reEdit()
curam.contracts.impl.FlatRateContract.FlatRateContractGenerateEvents	Raised when a flat rate contract is generated.	curam.contracts.impl.FlatRateContract.generate()
curam.contracts.impl.FlatRateContract.FlatRateContractTerminateEvents	Raised when a flat rate contract is terminated.	curam.contracts.impl.FlatRateContract.terminate()
curam.contracts.impl.FlatRateContract.FlatRateContractRenewEvents	Raised when a flat rate contract is renewed.	curam.contracts.impl.FlatRateContract.renew()
curam.contracts.impl.FlatRateContract.FlatRateContractCloneEvents	Raised when a flat rate contract is cloned.	curam.contracts.impl.FlatRateContract.cloneFlatRateCo

3.7.3 Utilization Contract Events

The following events are located in the curam.contracts.impl.UtilizationContract interface.

Table 38. Utilization Contract Event Details.

This table describes Utilization Contract Events

Event Class	Description	Event is raised before and after
UtilizationContractDeleteEvents	Raised when a Utilization Contract is deleted.	curam.contracts.impl.UtilizationContract.delete()
UtilizationContractGenerateEvents	Raised when a Utilization Contract is generated.	curam.contracts.impl.UtilizationContract.generate()
UtilizationContractActivateEvents	Raised when a Utilization Contract is activated	curam.contracts.impl.UtilizationContract.activate()

Table 38. Utilization Contract Event Details (continued).

This table describes Utilization Contract Events

Event Class	Description	Event is raised before and after
UtilizationContractTerminateEvents	Raised when a Utilization Contract is terminated.	curam.contracts.impl.UtilizationContract.terminate()
UtilizationContractRenewEvents	Raised when a Utilization Contract is renewed.	curam.contracts.impl.UtilizationContract.renew()
UtilizationContractReEditEvents	Raised when a Utilization Contract is edited.	curam.contracts.impl.UtilizationContract.reEdit()
UtilizationContractCloneUtilizationContractEvents	Raised when a Utilization Contract is cloned.	curam.contracts.impl.UtilizationContract.cloneUtilization()
UtilizationContractCloneUtilizationContractForRenewalEvents	Raised when a Utilization Contract is cloned for renewal.	curam.contracts.impl.UtilizationContract.cloneUtilization()
UtilizationContractAmendEvents	Raised when a Utilization Contract is amended	curam.contracts.impl.UtilizationContract.amend()

3.7.4 Performance Measure Events

The following events are located in the curam.performancemeasure.impl.RetrievePerformanceMeasureForContract interface.

Table 39. Performance Measure Event Details.

This table describes Performance Measure Events

Event Class	Description	Event is raised before and after
RetrievePerformanceMeasureForContractCustomEvents	Raised when the actual value for a custom Performance Measure is retrieved.	curam.performancemeasure.impl.RetrievePerformanceMe
RetrievePerformanceMeasureForContractAvgCostPerSuccessfulOutcomeEvents	Raised when an average cost per successful Outcome is retrieved.	curam.performancemeasure.impl.RetrievePerformanceMe
RetrievePerformanceMeasureForContractAvgCostPerUnitOfServiceEvents	Raised when the Performance Measure for an average cost per unit of service is retrieved.	curam.performancemeasure.impl.RetrievePerformanceMe
RetrievePerformanceMeasureForContractNoOfClientsServedEvents	Raised when the Performance Measure on number of clients served are retrieved.	curam.performancemeasure.impl.RetrievePerformanceMe
RetrievePerformanceMeasureForContractNoOfUnitsDeliveredEvents	Raised when the Performance Measure for number of units delivered are retrieved.	curam.performancemeasure.impl.RetrievePerformanceMe
RetrievePerformanceMeasureForContractRateOfAchievementOfSuccessfulOutcomeEvents	Raised when the Performance Measure for rate of achievement of successful outcome is retrieved.	curam.performancemeasure.impl.RetrievePerformanceMe
RetrievePerformanceMeasureForContractActualPerformanceEvents	Raised when the actual Performance Measure value is retrieved.	curam.performancemeasure.impl.RetrievePerformanceMe

3.8 Service Invoice Customization Points

The following sections list the available customization points for Service Invoices.

3.8.1 Service Invoice Events

The following events are located in the curam.financial.impl.ServiceInvoice interface.

Table 40. Service Invoice Event Details.

This table describes Service Invoice Events

Event Class	Description	Event is raised before and after
ServiceInvoiceAddLineItemEvents	Raised when a Service Invoice Line Item is added to a Service Invoice.	curam.financial.impl.ServiceInvoice.addLineItem()
ServiceInvoiceBulkApproveEvents	Raised when many Service Invoice Line Items are approved together in bulk.	curam.financial.impl.ServiceInvoice.bulkApprove()
ServiceInvoiceGetServiceInvoiceDerivedStatusEvents	Raised when a Service Invoice status is retrieved.	curam.financial.impl.ServiceInvoice.getServiceInvoiceD

3.8.2 Service Invoice Line Item Events

The following events are located in the curam.financial.impl.ServiceInvoiceLineItem interface.

Table 41. Service Invoice Line Item Event Details.

This table describes Service Invoice Line Item Events

Event Class	Description	Event is raised before and after
ServiceInvoiceLineItemApproveEvents	Raised when a Service Invoice Line Item is approved.	curam.financial.impl.ServiceInvoiceLineItem.approve()
ServiceInvoiceLineItemDenyEvents	Raised when a Service Invoice Line Item is denied.	curam.financial.impl.ServiceInvoiceLineItem.deny()
ServiceInvoiceLineItemSubmitEvents	Raised when a Service Invoice Line Item is submitted.	curam.financial.impl.ServiceInvoiceLineItem.submit()
ServiceInvoiceLineItemMatchCaseEvents	Raised when a Case Reference in a Service Invoice Line Item is matched with a case participant.	curam.financial.impl.ServiceInvoiceLineItem.matchCas
ServiceInvoiceLineItemMatchPayeeEvents	Raised when payee details on a Service Invoice Line Item are matched with a provider/provider group.	curam.financial.impl.ServiceInvoiceLineItem.matchPay
ServiceInvoiceLineItemMatchProviderEvents	Raised when the details of the provider who is providing the service as taken from the Service Invoice Line Item, are matched with a registered provider.	curam.financial.impl.ServiceInvoiceLineItem.matchPro
ServiceInvoiceLineItemMatchClientEvents	Raised when client details are matched with the client who received the service.	curam.financial.impl.ServiceInvoiceLineItem.matchCli
ServiceInvoiceLineItemResolveServiceAuthorizationEvents	Raised when a Service Authorization Line Item is matched to a Service Invoice Line Item.	curam.financial.impl.ServiceInvoiceLineItem.resolveSe
ServiceInvoiceLineItemValidateLineItemAgainstAuthorizationEvents	Raised when Service Authorization Line Item details are validated against Service Invoice Line Item details.	curam.financial.impl.ServiceInvoiceLineItem.validateL
ServiceInvoiceLineItemGeneratePaymentEvents	Raised when a payment is processed for a Service Invoice Line Item.	curam.financial.impl.ServiceInvoiceLineItem.generateP

Table 41. Service Invoice Line Item Event Details (continued).

This table describes Service Invoice Line Item Events

Event Class	Description	Event is raised before and after
ServiceInvoiceLineItemDeterminePaymentAmountEvent	Raised when the payment amount is determined from the established rates for the period specified in the Service Invoice Line Item.	curam.financial.impl.ServiceInvoiceLineItem.determinePa
ServiceInvoiceLineItemMatchIdentifiersEvent	Raised when Case, Provider, Client details on a Service Invoice Line Item are matched.	curam.financial.impl.ServiceInvoiceLineItem.matchIdenti
ServiceInvoiceLineItemDeterminePaymentAmountForReassessmentEvent	Raised when the payment amount is determined from the established rates to reassess the payment made for Service Invoice Line Item.	curam.financial.impl.ServiceInvoiceLineItem.determinePa
ServiceInvoiceLineItemMatchAgainstFlatRateContractEvent	Raised when Service Invoice Line Item details are matched with an existing Flat Rate Contract.	curam.financial.impl.ServiceInvoiceLineItem.matchAgain
ServiceInvoiceLineItemRetrieveServiceAuthorizationEvent	Raised when Service Authorization details related to a Service Invoice Line Item are retrieved.	curam.financial.impl.ServiceInvoiceLineItem.retrieveServi
ServiceInvoiceLineItemSubmitAndApproveSILIForCorrectionEvent	Raised when a Service Invoice Line Item Correction is submitted and approved.	curam.financial.impl.ServiceInvoiceLineItem.submitAndA
ServiceInvoiceLineItemRetrieveSILIAmountPaidEvent	Raised when the amount paid against a Service Invoice Line Item is retrieved.	curam.financial.impl.ServiceInvoiceLineItem.retrieveSILIA
ServiceInvoiceLineItemListSOAttendanceConfigurationEvent	Raised when the Service Offering Attendance Configuration for the Service Offering related to a Service Invoice Line Item is retrieved.	curam.financial.impl.ServiceInvoiceLineItem.listSOAttenc
ServiceInvoiceLineItemGetAmountPaidEvent	Raised when the amount paid/payable against a Service Invoice Line Item is retrieved.	curam.financial.impl.ServiceInvoiceLineItem.getAmountP

The following events are located in the curam.financial.impl.DeterminePaymentAmount interface.

Table 42. Service Invoice Line Item Event Details.

This table describes Service Invoice Line Item Events

Event Class	Description	Event is raised before and after
DeterminePaymentAmountDeterminePaymentAmountEvent	Raised when the amount to be paid for the Service Invoice Line Item is determined.	curam.financial.impl.DeterminePaymentAmount.determi

The following events are located in the curam.financial.impl.PaymentOptionProcessor interface.

Table 43. Service Invoice Line Item Event Details.

This table describes Service Invoice Line Item Events

Event Class	Description	Event is raised before and after
PaymentOptionProcessorProcessServiceEvent	Raised when the payment option changes for a Service Offering made through invoices for the specified reassessment period is processed.	curam.financial.impl.PaymentOptionProcessor.process
PaymentOptionProcessorProcessProviderEvent	Raised when the payment option changes for a Provider made through invoices for the specified reassessment period is processed.	curam.financial.impl.PaymentOptionProcessor.process

The following events are located in the curam.financial.impl.ServiceInvoiceLineItemHelper interface.

Table 44. Service Invoice Line Item Event Details.

This table describes Service Invoice Line Item Events

Event Class	Description	Event is raised before and after
ServiceInvoiceLineItemHelperMatchClientEvent	Raised when the client details with the client who received the service is matched.	curam.financial.impl.ServiceInvoiceLineItemHelper.ma
ServiceInvoiceLineItemHelperMatchProviderEvent	Raised when the Provider details with Provider/Provider Group who provided the service is matched.	curam.financial.impl.ServiceInvoiceLineItemHelper.ma
ServiceInvoiceLineItemHelperMatchCaseEvent	Raised when the case reference in Service Invoice Line Item to the participant case is matched.	curam.financial.impl.ServiceInvoiceLineItemHelper.ma
ServiceInvoiceLineItemHelperMatchPayeeEvent	Raised when the payee details with Provider/Provider Group is matched.	curam.financial.impl.ServiceInvoiceLineItemHelper.ma

The following events are located in the curam.financial.impl.ServiceInvoiceLineItemTransactionHelper interface.

Table 45. Service Invoice Line Item Event Details.

This table describes Service Invoice Line Item Events

Event Class	Description	Event is raised before and after
ServiceInvoiceLineItemTransactionHelperCreateCancelTransactionEvent	Raised when a Service Invoice Line Item transaction of type canceled is created.	curam.financial.impl.ServiceInvoiceLineItemTransactio
ServiceInvoiceLineItemTransactionHelperCreateDeniedTransactionEvent	Raised when a Service Invoice Line Item transaction of type denied is created.	curam.financial.impl.ServiceInvoiceLineItemTransactio
ServiceInvoiceLineItemTransactionHelperCreateInvoicedTransactionEvent	Raised when a new transaction with the type as Invoiced for a Service Invoice Line Item is created.	curam.financial.impl.ServiceInvoiceLineItemTransactio
ServiceInvoiceLineItemTransactionHelperCreatePaymentTransactionEvent	Raised when a new transaction with the type as Payment for a Service Invoice Line Item is created.	curam.financial.impl.ServiceInvoiceLineItemTransactio

3.8.3 Service Invoice Line Item Correction Events

The following events are located in the curam.financial.impl.ServiceInvoiceLineItemCorrection interface.

Table 46. Service Invoice Line Item Correction Event Details.

This table describes Service Invoice Line Item Correction Events

Event Class	Description	Event is raised before and after
ServiceInvoiceLineItemCorrectionApproveEvents	Raised when a Service Invoice Line Item Correction is approved.	curam.financial.impl.ServiceInvoiceLineItemCorrection.ap
ServiceInvoiceLineItemCorrectionDenyEvents	Raised when a Service Invoice Line Item Correction is denied.	curam.financial.impl.ServiceInvoiceLineItemCorrection.de
ServiceInvoiceLineItemCorrectionSubmitEvents	Raised when a Service Invoice Line Item Correction is submitted.	curam.financial.impl.ServiceInvoiceLineItemCorrection.su
ServiceInvoiceLineItemCorrectionValidateEvents	Raised when a Service Invoice Line Item is validated against the Service Authorization Line Item details.	curam.financial.impl.ServiceInvoiceLineItemCorrection.va

3.9 Attendance Customization Points

The following sections list the available customization points for Service Invoices.

3.9.1 Provider Roster Line Item Events

The following events are located in the curam.attendance.impl.ProviderRosterLineItem interface.

Table 47. Provider Roster Line Item Event Details.

This table describes Provider Roster Line Item Events

Event Class	Description	Event is raised before and after
ProviderRosterLineItemModifyRosterLineItemOnModificationOfSAEvents	Raised when a Roster Line Item is modified on modification of a service authorization line item.	curam.attendance.impl.ProviderRosterLineItem.modifyRo ()
ProviderRosterLineItemModifyRosterLineItemEvents	Raised when a Roster Line Item is modified.	curam.attendance.impl.ProviderRosterLineItem.modifyRo ()
ProviderRosterLineItemModifyForDailyAttendanceEvents	Raised when a Roster Line Item is modified based on daily attendance.	curam.attendance.impl.ProviderRosterLineItem.modifyFo
ProviderRosterLineItemApproveEvents	Raised when a Roster Line Item is approved.	curam.attendance.impl.ProviderRosterLineItem.approve()
ProviderRosterLineItemAddClientEvents	Raised when a Roster Line Item is created for a new client.	curam.attendance.impl.ProviderRosterLineItem.addClient
ProviderRosterLineItemAddAbsencePeriodEvents	Raised when an absence period is added to a Roster Line Item.	curam.attendance.impl.ProviderRosterLineItem.addAbsen
ProviderRosterLineItemSubmitEvents	Raised when a Roster Line Item is submitted.	curam.attendance.impl.ProviderRosterLineItem.submit()
ProviderRosterLineItemSubmitRosterLineItemFromRosterEvents	Raised when a Roster Line Item from a Roster is submitted.	curam.attendance.impl.ProviderRosterLineItem.submitRo
ProviderRosterLineItemDenyEvents	Raised when a Roster Line Item is denied.	curam.attendance.impl.ProviderRosterLineItem.deny()
ProviderRosterLineItemSubmitAndApproveRLIForCorrectionEvents	Raised when a Roster Line Item correction is submitted for approval.	curam.attendance.impl.ProviderRosterLineItem.submitAn

Table 47. Provider Roster Line Item Event Details (continued).

This table describes Provider Roster Line Item Events

Event Class	Description	Event is raised before and after
ProviderRosterLineItemAccommodateCreateOnExistingRoster	Raised when Roster Line Items accommodated on an existing Roster.	curam.attendance.impl.ProviderRosterLineItem.accom
ProviderRosterLineItemCalculateExpectedUnits	Raised when expected units on a Provider Roster Line Item are calculated.	curam.attendance.impl.ProviderRosterLineItem.calcula
ProviderRosterLineItemUpdateExpectedUnitsOnNextRoster	Raised when expected units for Roster Line Items on a Roster for a particular client are updated.	curam.attendance.impl.ProviderRosterLineItem.update
ProviderRosterLineItemListSOAttendanceConfigurationForServiceOffering	Raised when the SO Attendance Configuration list for a Roster Line Item is retrieved.	curam.attendance.impl.ProviderRosterLineItem.listSO
ProviderRosterLineItemViewExceptionTaskEvent	Raised when an exception task is viewed for a Provider Roster Line Item.	curam.attendance.impl.ProviderRosterLineItem.viewE
ProviderRosterLineItemGetCorrectionIndicator	Raised when the correction indicator for a Provider Roster Line Item is retrieved.	curam.attendance.impl.ProviderRosterLineItem.getCor
ProviderRosterLineItemGetCaseHeaderDetailsEvent	Raised when Case Header Details are retrieved.	curam.attendance.impl.ProviderRosterLineItem.getCas
ProviderRosterLineItemGetPayBasedOnAttendanceIndicator	Raised when the Pay Based on Attendance is retrieved.	curam.attendance.impl.ProviderRosterLineItem.getPay
ProviderRosterLineItemGetAbsencePeriodDetails	Raised when the Absence period on a Provider Roster Line Item is retrieved.	curam.attendance.impl.ProviderRosterLineItem.getAbs
ProviderRosterLineItemGetDailyAttendanceDetails	Raised when Daily attendance is retrieved from a Provider Roster Line Item.	curam.attendance.impl.ProviderRosterLineItem.getDai
ProviderRosterLineItemGetOriginalDtlsEvent	Raised when Provider Roster Line Item details are retrieved.	curam.attendance.impl.ProviderRosterLineItem.getOri

The following events are located in the curam.attendance.impl.AttendanceInformationProcessing interface.

Table 48. Provider Roster Line Item Event Details.

This table describes Provider Roster Line Item Events

Event Class	Description	Event is raised before and after
AttendanceInfoProcessGetRosterLineItemsForCaseEvents	Raised when Provider Roster Line Items for a case is retrieved.	curam.attendance.impl.AttendanceInformationProcess
AttendanceInfoProcessGetRosterLineItemsForClientEvents	Raised when Provider Roster Line Items for a client is retrieved.	curam.attendance.impl.AttendanceInformationProcess

The following events are located in the curam.attendance.impl.ProviderRosterLineItemHelper interface.

Table 49. Provider Roster Line Item Event Details.

This table describes Provider Roster Line Item Events

Event Class	Description	Event is raised before and after
ProviderRosterLineItemHelperMatchClientEvents	Raised when the client is matched based on client reference number, first name and last name and address.	curam.attendance.impl.ProviderRosterLineItemHelper.ma
ProviderRosterLineItemHelperMatchCaseEvents	Raised when the case is matched by the case reference number.	curam.attendance.impl.ProviderRosterLineItemHelper.ma
ProviderRosterLineItemHelperMatchVoucherEvents	Raised when the voucher is matched by number assigned to the voucher that has been issued to the client.	curam.attendance.impl.ProviderRosterLineItemHelper.ma

The following events are located in the curam.attendance.impl.ProviderRosterLineItemTransactionHelper interface.

Table 50. Provider Roster Line Item Event Details.

This table describes Provider Roster Line Item Events

Event Class	Description	Event is raised before and after
ProviderRosterLineItemTransactionHelperCreateDenialTransactionEvents	Raised when a Provider Roster Line Item transaction of type denied is created.	curam.attendance.impl.ProviderRosterLineItemTransactio
ProviderRosterLineItemTransactionHelperCreateCancelTransactionEvents	Raised when a Provider Roster Line Item transaction of type canceled is created.	curam.attendance.impl.ProviderRosterLineItemTransactio
ProviderRosterLineItemTransactionHelperCreateProvidedTransactionEvents	Raised when Provider Roster Line Item transactions are created.	curam.attendance.impl.ProviderRosterLineItemTransactio

3.9.2 Provider Roster Line Item Correction (PRLI Correction) Events

The following events are located in the curam.attendance.impl.PRLICorrection interface.

Table 51. Provider Roster Line Item Correction (PRLI Correction) Event Details.

This table describes Provider Roster Line Item Correction (PRLI Correction) Events

Event Class	Description	Event is raised before and after
PRLICorrectionApproveEvents	Raised when a Provider Roster Line Item Correction is approved.	curam.attendance.impl.PRLICorrection.approve()
PRLICorrectionDenyEvents	Raised when a Provider Roster Line Item Correction is denied.	curam.attendance.impl.PRLICorrection.deny()
PRLICorrectionSubmitEvents	Raised when a Provider Roster Line Item Correction is submitted.	curam.attendance.impl.PRLICorrection.submit()

3.9.3 Roster Events

The following events are located in the curam.attendance.impl.RosterProcessing interface.

Table 52. Roster Event Details.

This table describes Roster Events

Event Class	Description	Event is raised before and after
RosterProcessingGenerateRosterManually Events	Raised when blank roster is generated manually.	curam.attendance.impl.RosterProcessing.generateRoste
RosterProcessingGetApplicableRosterRange Events	Raised when the applicable roster range is retrieved.	curam.attendance.impl.RosterProcessing.getApplicable
RosterProcessingCreateRosterOverlappingDate Events	Raised when roster for Service Authorization Line Item overlapping date is created.	curam.attendance.impl.RosterProcessing.createRosterC
RosterProcessingCreateRosterEvents	Raised when roster for a Service Authorization Line Item is created.	curam.attendance.impl.RosterProcessing.createRoster()

3.9.4 Attendance Payment Frequency Events

The following events are located in the curam.attendance.impl.AttendancePaymentFrequency interface.

Table 53. Attendance Payment Frequency Event Details.

This table describes Attendance Payment Frequency Events

Event Class	Description	Event is raised before and after
AttendancePaymentFrequencyGetDerivedStatus Events	Raised when the status of an attendance payment configuration entry is retrieved.	curam.attendance.impl.AttendancePaymentFrequency. ()

3.9.5 Service Offering Attendance Configuration Events

The following events are located in the curam.attendance.impl.SOAttendanceConfiguration interface.

Table 54. Service Offering Attendance Configuration Event Details.

This table describes Service Offering Attendance Configuration Events

Event Class	Description	Event is raised before and after
SOAttendanceConfigurationGetDerivedStatus Events	Raised when the status of a Service Offering Attendance Configuration is retrieved.	curam.attendance.impl.SOAttendanceConfiguration.ge

3.9.6 Service Offering Attendance Payment Events

The following events are located in the interface.

Table 55. Service Offering Attendance Event Details.

This table describes Service Offering Attendance Payment

Event Class	Description	Event is raised before and after
SOAttendancePaymentGetDerivedStatus Events	Raised when the status of a Service Offering Attendance Payment is retrieved.	curam.attendance.impl.SOAttendancePayment.getDeri

3.10 Financial Customization Points

The following sections list the available customization points for Financials.

3.10.1 Financial Events

The following events are located in the curam.financial.impl.FinancialAPI interface.

Table 56. Financial Event Details.

This table describes Financial Events

Event Class	Description	Event is raised before and after
FinancialAPIRetrieveServiceDeliverySummaryInformationEvents	Raised when the service delivery summary information is retrieved for a a case, client and service.	curam.financial.impl.FinancialAPI.retrieveServiceDelivery
FinancialAPIRetrieveServiceDelivrySummaryInformationEvents	Raised when the service delivery summary information is retrieved for a service and case participant role.	curam.financial.impl.FinancialAPI.retrieveServiceDelivery
FinancialAPIListReassessmentResultsEvents	Raised when the reassessment results for all the product deliveries created to deliver the services for the given service authorization is retrieved.	curam.financial.impl.FinancialAPI.listReassessmentResult

The following events are located in the curam.financial.impl.GenerateOverUnderPayment interface.

Table 57. Financial Event Details.

This table describes Financial Events

Event Class	Description	Event is raised before and after
GenerateOverUnderPaymentGenerateOverPaymentForPRLIEvents	Raised when the Over payment for Provider Roster Line Item is generated.	curam.financial.impl.GenerateOverUnderPayment.genera
GenerateOverUnderPaymentGenerateOverPaymentForSILIEvents	Raised when the Over payment for Service Invoice Line Item is generated.	curam.financial.impl.GenerateOverUnderPayment.genera
GenerateOverUnderPaymentGenerateUnderPaymentForPRLIEvents	Raised when the Under payment for Provider Roster Line Item is generated.	curam.financial.impl.GenerateOverUnderPayment.genera
GenerateOverUnderPaymentGenerateUnderPaymentForSILIEvents	Raised when the Under payment for Service Invoice Line Item is generated.	curam.financial.impl.GenerateOverUnderPayment.genera

The following events are located in the curam.financial.impl.PaymentProcessing interface.

Table 58. Financial Event Details.

This table describes Financial Events

Event Class	Description	Event is raised before and after
PaymentProcessingProcessPaymentForSILIEvents	Raised when the payment for the Service Invoice Line Item is processed.	curam.financial.impl.PaymentProcessing.processPayment
PaymentProcessingProcessPaymentForReassessmentEvents	Raised when the payment for reassessment is processed.	curam.financial.impl.PaymentProcessing.processPayment

Table 58. Financial Event Details (continued).

This table describes Financial Events

Event Class	Description	Event is raised before and after
PaymentProcessingApproveAndActivateCaseEvents	Raised when the cases for Provider is approved and activated.	curam.financial.impl.PaymentProcessing.approveAndA
PaymentProcessingSubmitForApprovalEvents	Raised when the case is submitted and approved.	curam.financial.impl.PaymentProcessing.submitForAp
PaymentProcessingDeterminePayeeDetailsEvents	Raised when the payee for a given Provider and given period is determined.	curam.financial.impl.PaymentProcessing.determinePay
PaymentProcessingDeterminePayeDetailsEvents	Raised when the payee for a given Provider and given period is determined.	curam.financial.impl.PaymentProcessing.determinePay

The following events are located in the curam.financial.impl.ProcessCaseNominee interface.

Table 59. Financial Event Details.

This table describes Financial Events

Event Class	Description	Event is raised before and after
ProcessCaseNomineeCreateCaseNomineeEvents	Raised when the Provider Group is created as the case nominee for the given product delivery case of the Provider.	curam.financial.impl.ProcessCaseNominee.createCaseN
ProcessCaseNomineeCreateCaseNomineeEvents	Raised when the payee is created as the case nominee for the given product delivery case of the Provider.	curam.financial.impl.ProcessCaseNominee.createCaseN
ProcessCaseNomineeCreateCaseNomineeEvents	Raised when the Provider Group is created as the case nominee for the given product delivery case of the Provider and contract frequency.	curam.financial.impl.ProcessCaseNominee.createCaseN
ProcessCaseNomineeCreateNomineeForAllCasesEvents	Raised when the case nominees are created for all existing cases associated with the Provider for whom the Provider Group Associate Payment Configuration is created.	curam.financial.impl.ProcessCaseNominee.createNomi
ProcessCaseNomineeReassignCaseNomineeObjectivesOnCancellationEvents	Raised when the case nominee objectives associated with the Provider Group Associate Payment Configuration is reassigned on cancellation of payment configuration.	curam.financial.impl.ProcessCaseNominee.reassignCas
ProcessCaseNomineeReassignCaseNomineeObjectivesOnModificationEvents	Raised when the case nominee objectives associated with the Provider Group Associate Payment Configuration is reassigned on modification of Provider Group Associate.	curam.financial.impl.ProcessCaseNominee.reassignCas

Table 59. Financial Event Details (continued).

This table describes Financial Events

Event Class	Description	Event is raised before and after
ProcessCaseNomineeReassignCaseNomineeEvents	Raised when the Objective Modification Events objectives associated with the Provider Group Associate Payment Configuration is reassigned on modification of payment configuration.	curam.financial.impl.ProcessCaseNominee.reassignCaseN
ProcessCaseNomineeReAssignCaseNomineeEvents	Raised when the Objective Modification Events objectives associated with the old payee to the new payee for the given product delivery case is reassigned.	curam.financial.impl.ProcessCaseNominee.reAssignCaseN

The following events are located in the curam.financial.impl.RateValidator interface.

Table 60. Financial Event Details.

This table describes Financial Events

Event Class	Description	Event is raised before and after
RateValidatorValidateRatesEvents	Raised when there is any gap or overlapping in the period of the set of rates provided are validated.	curam.financial.impl.RateValidator.validateRates()

3.11 Referral Customization Points

The following sections list the available customization points for Referrals.

3.11.1 Referral Events

The following events are located in the curam.referral.impl.Referral interface.

Table 61. Referral Event Details.

This table describes Referral Events

Event Class	Description	Event is raised before and after
ReferralSendNotificationEvents	Raised when a notification letter to the Concern Role is sent.	curam.referral.impl.Referral.sendNotification()
ReferralCreateReferralRoleEvents	Raised when a referral role record for a referral is created.	curam.referral.impl.Referral.createReferralRole()

The following events are located in the curam.referral.impl.ReferralNotification interface.

Table 62. Referral Event Details.

This table describes Referral Events

Event Class	Description	Event is raised before and after
ReferralNotificationGenerateNotificationEvents	Raised when a notification document is generated.	curam.referral.impl.ReferralNotification.generateNotificat
ReferralNotificationSendNotificationEvents	Raised when a notification is sent to the Concern Role.	curam.referral.impl.ReferralNotification.sendNotification(

3.12 Service Delivery Customization Points

The following sections list the available customization points for Service Deliveries.

3.12.1 Service Delivery Events

The following events are located in the curam.servicedelivery.impl.ServiceDeliveryEstimatedCost interface.

Table 63. Service Delivery Event Details.

This table describes Service Delivery Events

Event Class	Description	Event is raised before and after
ServiceDeliveryEstimatedCostDetermineRateEvents	Raised when the rate for the Service Offering is determined.	curam.servicedelivery.impl.ServiceDeliveryEstimatedC
ServiceDeliveryEstimatedCostDetermineRateWithEffectiveDateEvents	Raised when the rate for the Service Offering is determined for each service occurrence date.	curam.servicedelivery.impl.ServiceDeliveryEstimatedC

The following events are located in the curam.servicedelivery.impl.ServiceDelivery interface.

Table 64. Service Delivery Event Details.

This table describes Service Delivery Events

Event Class	Description	Event is raised before and after
ServiceDeliverySubmitEvents	Raised when the Service Delivery is submitted.	curam.servicedelivery.impl.ServiceDelivery.submit()

The following events are located in the curam.servicedeliveryevaluation.impl.ServiceDeliveryEvaluation interface.

Table 65. Service Delivery Event Details.

This table describes Service Delivery Events

Event Class	Description	Event is raised before and after
ServiceDeliveryEvalCalculateOutcomeForServiceDeliveryEvaluationEvents	Raised when the outcome for a Service Delivery Evaluation is calculated.	curam.servicedeliveryevaluation.impl.ServiceDeliveryE

3.13 Taxonomy Customization Points

The following sections list the available customization points for Taxonomy.

3.13.1 Taxonomy Events

The following events are located in the curam.taxonomy.impl.POTaxonomyInEditIndexProcessor interface.

Table 66. Taxonomy Event Details.

This table describes Taxonomy Events

Event Class	Description	Event is raised before and after
POTaxonomyInEditIndexProcessorPublishInEditIndexEvent	Raised when the updated Provider Offering and Taxonomy indexing details are published.	curam.taxonomy.impl.POTaxonomyInEditIndexProcessor
POTaxonomyInEditIndexProcessorAddInEditTermDataEvent	Raised when the updated term data to the Provider Offering and Taxonomy Term association details are added.	curam.taxonomy.impl.POTaxonomyInEditIndexProcessor

The following events are located in the curam.taxonomy.impl.PublishTaxonomy interface.

Table 67. Taxonomy Event Details.

This table describes Taxonomy Events

Event Class	Description	Event is raised before and after
PublishTaxonomyPublishInEditTaxonomyDataEvents	Raised when the In Edit Taxonomy Terms and its associated data are published.	curam.taxonomy.impl.PublishTaxonomy.publishInEditTax
PublishTaxonomyPublishInEditTaxonomyTermEvents	Raised when the In Edit Taxonomy Terms are published.	curam.taxonomy.impl.PublishTaxonomy.publishInEditTax
PublishTaxonomyPublishTaxonomyEvents	Raised when the Taxonomy Version is published.	curam.taxonomy.impl.PublishTaxonomy.publishTaxonom
PublishTaxonomyRemoveTaxonomyVersionEvents	Raised when the Taxonomy Version is removed.	curam.taxonomy.impl.PublishTaxonomy.removeTaxonom

The following events are located in the curam.taxonomy.impl.RelatedConceptWizState interface.

Table 68. Taxonomy Event Details.

This table describes Taxonomy Events

Event Class	Description	Event is raised before and after
RelatedConceptWizStateReadTaxonomyTerms	Raised when the Taxonomy Term list is retrieved from the wizard state.	curam.taxonomy.impl.RelatedConceptWizState.readTaxon
RelatedConceptWizStateRemoveInEditTerm	Raised when the In Edit Taxonomy Term is removed from the wizard state.	curam.taxonomy.impl.RelatedConceptWizState.removeIn
RelatedConceptWizStateRemoveTaxonomyTerm	Raised when the Taxonomy Terms are removed from the wizard state.	curam.taxonomy.impl.RelatedConceptWizState.removeTa
RelatedConceptWizStateResetSearch	Raised when the search criteria and Taxonomy Terms stored in the wizard state is reset.	curam.taxonomy.impl.RelatedConceptWizState.resetSearc
RelatedConceptWizStateSaveRelatedConcept	Raised when the Related Concept is added.	curam.taxonomy.impl.RelatedConceptWizState.saveRelate
RelatedConceptWizStateSaveRelatedConceptElement	Raised when the In Edit Related Concept is added.	curam.taxonomy.impl.RelatedConceptWizState.saveRelate

Table 68. Taxonomy Event Details (continued).

This table describes Taxonomy Events

Event Class	Description	Event is raised before and after
RelatedConceptWizStateStoreRelatedConcept	Raised when the Related Concept details are stored in the wizard state.	curam.taxonomy.impl.RelatedConceptWizState.storeRe
RelatedConceptWizStateStoreTaxonomyTerms	Raised when the Taxonomy Terms are stored in the wizard state.	curam.taxonomy.impl.RelatedConceptWizState.storeTa

The following events are located in the curam.taxonomy.impl.ReviewTaxonomy interface.

Table 69. Taxonomy Event Details.

This table describes Taxonomy Events

Event Class	Description	Event is raised before and after
ReviewTaxonomyAcceptModifiedTermEvent	Raised when the changes on the term based on review attribute selects are accepted.	curam.taxonomy.impl.ReviewTaxonomy.acceptModifie
ReviewTaxonomyAcceptReplacedTermEvent	Raised when the new terms selected for the replacement are accepted.	curam.taxonomy.impl.ReviewTaxonomy.acceptReplace
ReviewTaxonomyDeleteReplacedTermEvent	Raised when the term which is being replaced is deleted.	curam.taxonomy.impl.ReviewTaxonomy.deleteReplace
ReviewTaxonomyRejectReplacedTermEvent	Raised when the new replacement terms are rejected.	curam.taxonomy.impl.ReviewTaxonomy.rejectReplaced
ReviewTaxonomyRejectModifiedTermEvent	Raised when the changes in the term is rejected.	curam.taxonomy.impl.ReviewTaxonomy.rejectModified
ReviewTaxonomyReviewDeletedTermsEvent	Raised when the deleted terms are reviewed.	curam.taxonomy.impl.ReviewTaxonomy.reviewDeleted
ReviewTaxonomyReviewAllDeletedTermsEvent	Raised when all the deleted terms are reviewed.	curam.taxonomy.impl.ReviewTaxonomy.reviewAllDele

The following events are located in the curam.taxonomy.impl.TaxonomyInEditData interface.

Table 70. Taxonomy Event Details.

This table describes Taxonomy Events

Event Class	Description	Event is raised before and after
TaxonomyInEditDataAddTermNameTranslation	Raised when the new translations are added to Taxonomy Term name attribute.	curam.taxonomy.impl.TaxonomyInEditData.addTermN
TaxonomyInEditDataAddTermBibliographicReferenceTranslation	Raised when the new translations are added to Taxonomy Term bibliographic reference attribute.	curam.taxonomy.impl.TaxonomyInEditData.addTermB
TaxonomyInEditDataAddTermCommentsTranslation	Raised when the new translations are added to Taxonomy Term comments attribute.	curam.taxonomy.impl.TaxonomyInEditData.addTermC
TaxonomyInEditDataAddTermDefinitionTranslation	Raised when the new translations are added to Taxonomy Term definition attribute.	curam.taxonomy.impl.TaxonomyInEditData.addTermD
TaxonomyInEditDataAddUseReferenceTranslation	Raised when the new translations are added to Use Reference.	curam.taxonomy.impl.TaxonomyInEditData.addUseRe

Table 70. Taxonomy Event Details (continued).

This table describes Taxonomy Events

Event Class	Description	Event is raised before and after
TaxonomyInEditDataAssociateTermWithRelatedConcepts	Raised when the term with Related Concepts are associated.	curam.taxonomy.impl.TaxonomyInEditData.associateTerm
TaxonomyInEditDataRemoveRelatedConceptAssociation	Raised when the association between In Edit Taxonomy Term from the Related Concept is removed.	curam.taxonomy.impl.TaxonomyInEditData.removeRelate

The following events are located in the curam.taxonomy.impl.TaxonomyTermRelatedConcept interface.

Table 71. Taxonomy Event Details.

This table describes Taxonomy Events

Event Class	Description	Event is raised before and after
TaxonomyTermRelatedConceptInsertRelatedConcepts	Raised when Related Concepts are added to a Taxonomy Term.	curam.taxonomy.impl.TaxonomyTermRelatedConcept.ins
TaxonomyTermRelatedConceptInsertTaxonomyTerms	Raised when Taxonomy Terms are added to a Related Concept.	curam.taxonomy.impl.TaxonomyTermRelatedConcept.ins

The following events are located in the curam.taxonomy.impl.TaxonomyTermWizState interface.

Table 72. Taxonomy Event Details.

This table describes Taxonomy Events

Event Class	Description	Event is raised before and after
TaxonomyTermWizStateReadRelatedConcepts	Raised when the Related Concept list is retrieved from the wizard state.	curam.taxonomy.impl.TaxonomyTermWizState.readRelate
TaxonomyTermWizStateReadRelatedTerms	Raised when the related Taxonomy Term list is retrieved from the wizard state.	curam.taxonomy.impl.TaxonomyTermWizState.readRelate
TaxonomyTermWizStateReadUseReferences	Raised when the Use Reference list is retrieved from the wizard state.	curam.taxonomy.impl.TaxonomyTermWizState.readUseRe
TaxonomyTermWizStateRemoveRelatedTerm	Raised when the related term is removed from the related term list in the wizard state.	curam.taxonomy.impl.TaxonomyTermWizState.removeRe
TaxonomyTermWizStateRemoveUseReference	Raised when an Use Reference is removed from the Use Reference list in the wizard state.	curam.taxonomy.impl.TaxonomyTermWizState.removeUs
TaxonomyTermWizStateSaveAll	Raised when the Taxonomy Term and Use References are created from the details in the wizard state.	curam.taxonomy.impl.TaxonomyTermWizState.saveAll()
TaxonomyTermWizStateStoreRelatedConcept	Raised when the Related Concept is stored in the data store.	curam.taxonomy.impl.TaxonomyTermWizState.storeRelat
TaxonomyTermWizStateStoreRelatedTerms	Raised when the related terms are stored in the wizard state.	curam.taxonomy.impl.TaxonomyTermWizState.storeRelat
TaxonomyTermWizStateStoreTaxonomyTerm	Raised when the Taxonomy Term details are stored in the wizard state.	curam.taxonomy.impl.TaxonomyTermWizState.storeTaxon
TaxonomyTermWizStateStoreUseReference	Raised when the Use Reference is stored in the wizard state.	curam.taxonomy.impl.TaxonomyTermWizState.storeUseR

The following events are located in the curam.taxonomy.sl.search.impl.POTaxonomyIndexSearch interface.

Table 73. Taxonomy Event Details.

This table describes Taxonomy Events

Event Class	Description	Event is raised before and after
POTaxonomyIndexSearchIsProviderOfferingAlreadyIndexedEvents	Raised when the event value which represents whether Provider Offering is indexed or not is retrieved.	curam.taxonomy.sl.search.impl.POTaxonomyIndexSear
POTaxonomyIndexSearchIsSameServiceAlreadyIndexedInOtherProviderEvents	Raised when the event value whether the same service is already indexed in other Providers with different or same Taxonomy Terms is retrieved.	curam.taxonomy.sl.search.impl.POTaxonomyIndexSear

Chapter 4. CPM Workflow Process Definitions

4.1 Introduction

CPM ships a number of workflow process definitions. Agencies can copy any of these workflow process definitions to a custom workflow directory and make modifications to them.

Note: Custom versions of workflows will always take precedence over OOTB workflows.

4.2 External Enquiry Workflow

4.2.1 Enacted from

This workflow is enacted when an external party uses CPM to inquire about the possibility of registering as a provider. For example, Mr and Mrs Smith use an external-facing system to inquire about fostering children.This workflow is enacted by curam.cpm.eua.facade.impl.ExternalProviderEnquiry.createEnquiry.

4.2.2 Source Location

EJBServer/components/CPM/workflow/EXTERNALENQUIRYWORKFLOW_v1.xml

4.2.3 Default Behavior

The workflow shipped with CPM creates a manual activity to assign the enquiry to a user for converting a provider enquiry into an enrolled provider. This manual activity is allocated using a function allocation strategy. The default implementation of this operation allocates the activity to the provider enquiry work queue. The reviewer can choose to either transfer the enquiry to an enrolled provider or close the enquiry. Once this activity is completed the workflow also gets completed.

4.2.4 Event Details

The notation of the following event details is as follows:

Table 74. External Enquiry Event Details

Event Raised	Primary Event Data	Raised From
PROVIDERENQUIRY.TRANSFERENQUIRYTOPROVIDER	providerEnquiryID	curam.cpm.facade.impl.ProviderEnquiry.closeProviderEnquiry
PROVIDERENQUIRY.CLOSEENQUIRY	providerEnquiryID	curam.cpm.facade.impl.ProviderEnquiry.transferEnquiryToPro

4.3 Home Study Approval Workflow

4.3.1 Enacted from

This workflow is enacted whenever a user submits a home study for approval.This workflow is enacted from curam.cpm.workflowprocesses.homestudy.impl.HomeStudyImpl.submit.

4.3.2 Source Location

EJBServer/components/CPM/workflow/HOMESTUDYAPPROVAL_v1.xml

4.3.3 Default Behavior

This workflow automatically creates a manual activity to assign a home study recommendation to a user for approval. The default implementation of this operation submits the home study recommendation to the supervisor of the user who submitted the request. Agencies may wish to alter this default behavior.

For example, an agency may wish to route the approval request to a user other than the supervisor or to a group of users. The manual activity is allocated using a function allocation strategy. The reviewer can choose to either approve/reject the approval request. Once this activity is completed the workflow also gets completed.

4.3.4 Event Details

The notation of the following event details is as follows:

Table 75. Home Study Approval Event Details

Event Raised	Primary Event Data	Raised From
PROVIDERMANAGEMENT.HOMESTUDYAPPROVED	homeStudyID	curam.homestudy.impl.approve
PROVIDERMANAGEMENT.HOMESTUDYRETURNED	homeStudyID	curam.homestudy.impl.reject

4.4 New Invoice Created Workflow

4.4.1 Enacted from

This workflow is enacted whenever an external user submits an invoice for processing.This workflow is enacted from curam.cpm.facade.impl.Request.createFinancialsTask.

4.4.2 Source Location

EJBServer/components/CPM/workflow/NEWINVOICECREATED_v1.xml

4.4.3 Default Behavior

This workflow creates a manual activity to assign an invoice that has been submitted by an external user to another user for processing. The default implementation of this operation submits the invoice to a financial user. Agencies may wish to alter this default behavior. For example, an agency may wish to submit the invoice to a different user for processing. The manual activity is allocated using a function allocation strategy.

4.4.4 Event Details

The notation of the following event details is as follows:

Table 76. New Invoice Created Event Details

Event Raised	Primary Event Data	Raised From
NEWINVOICECREATED.INVOICECANCELLED	serviceInvoiceID	curam.cpm.facade.impl.cancelServiceInvoice
NEWINVOICECREATED.INVOICESUBMITTED	serviceInvoiceID	curam.cpm.facade.impl.submitSILIForProcessing

4.5 Service Invoice Exception Processing Workflow

4.5.1 Enacted from

This workflow is enacted when there is insufficient correct data to match a service invoice line item against its corresponding service authorization. This workflow is called from curam.financial.impl.ServiceInvoiceLineItemImpl.processInvoiceLineItem.

4.5.2 Source Location

EJBServer/components/CPM/workflow/SERVICEINVOICEEXCEPTIONPROCESSING_v1.xml.

4.5.3 Default Behavior

This workflow creates a manual activity for a user to review service invoice details that do not correspond with the service authorization associated with the invoice. During invoice processing, certain details on a service invoice line item (SILI) must correspond to the details on the service authorization that is associated with the invoice otherwise the invoice will not be paid. The default implementation of this operation allocates the activity to the invoice exception processing work queue for a financial user to review. Agencies may wish to alter this default behavior, for example, by routing the activity to a different work queue. The reviewer can choose to make changes to the service invoice line item and submit for reevaluation or deny/cancel the SILI. Once this activity is completed, the workflow is also completed. This manual activity is allocated using a function allocation strategy. The modeled operation for this is:
curam.cpm.workflowprocesses.impl.WorkflowAllocationFunction.siliExceptionProcessingAllocationStrategy.

4.5.4 Event Details

The notation of the following event details is as follows:

Table 77. Service Invoice Exception Processing Event Details

Event Raised	Primary Event Data	Raised From
PROVIDERMANAGEMENT.SILIPROCESSED	serviceInvoiceLineItemID	curam.financial.impl.ServiceInvoiceLineItemImpl.subm
PROVIDERMANAGEMENT.SILICANCELLED	serviceInvoiceLineItemID	curam.financial.impl.ServiceInvoiceLineItemImpl.canc
PROVIDERMANAGEMENT.SILIDENIED	serviceInvoiceLineItemID	curam.financial.impl.ServiceInvoiceLineItemImpl.deny

4.6 Service Invoice Line Item Approval Workflow

4.6.1 Enacted from

This workflow is enacted when a service invoice line item requires manual approval and has reached the "Pending Approval" status, after successful processing. This workflow is called from curam.financial.impl.ServiceInvoiceLineItemImpl.enactSILIApprovalWorkflow.

4.6.2 Source Location

EJBServer/components/CPM/workflow/SERVICEINVOICELINEITEMAPPROVAL_v1.xml.

4.6.3 Default Behavior

The workflow creates a manual activity to review a service invoice line item and approve/deny it. The default implementation of this operation allocates the activity to the invoice exception processing work queue for a financial user to approve. Agencies may wish to alter this default behavior, for example, by routing the activity to a different work queue or to a different user. This manual activity is allocated using a function allocation strategy. The modeled operation for this is curam.cpm.workflowprocesses.impl.WorkflowAllocationFunction.siliExceptionProcessingAllocationStrategy.

The reviewer can choose to either approve or deny the service invoice line item. Once this activity is completed the workflow also gets completed.

4.6.4 Event Details

The notation of the following event details is as follows:

Table 78. Service Invoice Line Item Event Details

Event Raised	Primary Event Data	Raised From
PROVIDERMANAGEMENT.SILIAPPROVED	serviceInvoiceLineItemID	curam.financial.impl.ServiceInvoiceLineItemImpl.approve
PROVIDERMANAGEMENT.SILIDENIED	serviceInvoiceLineItemID	curam.financial.impl.ServiceInvoiceLineItemImpl.deny

4.7 Service Invoice Line Item Correction Approval Workflow

4.7.1 Enacted From

This workflow is enacted when a service invoice line item correction is submitted for approval. This workflow is called from curam.financial.impl.ServiceInvoiceLineItemCorrectionImpl.enactCorrectionApprovalWorkflow.

4.7.2 Source Location

EJBServer/components/CPM/workflow/SERVICEINVOICELINEITEMCORRECTIONAPPROVAL_v1.xml

4.7.3 Default Behavior

This workflow automatically creates a manual activity to assign a service invoice line item correction to a user for approval. This manual activity is allocated using a function allocation strategy. The modeled operation for this curam.cpm.workflowprocesses.impl.WorkflowAllocationFunction.siliExceptionProcessingAllocationStrategy. The default implementation of this operation submits the service invoice line item correction to the supervisor of the user who submitted the request. Agencies may wish to alter this default behavior. For example, an agency may wish to route the approval request to a user other than the supervisor or to a group of users. The reviewer can choose to either approve or deny the service invoice line item correction. Once this activity is completed, the workflow is also completed.

4.7.4 Event Details

The notation of the following event details is as follows:

Table 79. Service Invoice Line Item Correction Approval Event Details

Event Raised	Primary Event Data	Raised From
PROVIDERMANAGEMENT.SILICORRECTIONAPPROVED	serviceInvoiceLineItemID	curam.financial.impl.ServiceInvoiceLineItemCorrectionImpl.appr
PROVIDERMANAGEMENT.SILICORRECTIONDENIED	serviceInvoiceLineItemID	curam.financial.impl.ServiceInvoiceLineItemCorrectionImpl.deny

4.8 Supervisor Request Decision Workflow

4.8.1 Enacted From

This workflow is enacted when a user submits a request to be set up with an external user account. This workflow is called from curam.cpm.eua.facade.impl.ExternalRequests.submitRequest.

4.8.2 Source Location

EJBServer/components/CPM/workflow/SUPERVISORREQUESTDECISION_v1.xml

4.8.3 Default Behavior

This workflow creates a manual activity that submits a request of an external user to an administrator user, who is set up as one of the following:

- A provider member
- A provider participant
- A provider group member
- A provider group associate

The default implementation submits the request to the external request work queue for an administrator to approve. Agencies may wish to alter this default behavior, for example, by routing the activity to a different work queue. The administrator can approve or reject the request of the external user. Once this activity is completed, the workflow is also completed.

4.8.4 Event Details

The notation of the following event details is as follows:

Table 80. Supervisor Request Decision Event Details

Event Raised	Primary Event Data	Raised From
REQUESTDECISION.REQUESTACCEPTED	requestID	curam.cpm.facade.impl.Request.raiseAcceptRequestEvent
REQUESTDECISION.REQUESTREJECTED	requestID	curam.cpm.facade.impl.Request.rejectRequest

4.9 Supervisor View New External User Task Notification Workflow

4.9.1 Enacted From

This workflow is enacted when an administrator user creates an external user account. This workflow is called from curam.cpm.eua.facade.impl.ExternalUser.createExternalUser.

4.9.2 Source Location

EJBServer/components/CPM/workflow/
SUPERVISORVIEWNEWEXTERNALUSERTASKNOTIFICATION_v1.xml

4.9.3 Default Behavior

This workflow creates a route activity to send a notification to the owner of an external user. When an administrator user sets up a new external user account, the owner of the new external user is sent a notification informing them that the account has been successfully created. The default implementation sends a notification to the resource manager who enrolled the external user. Agencies may wish to alter this default. For example, an agency may wish to send the notification to a different user. Once this activity is completed, the workflow is also completed.

4.9.4 Events Details

No Events are raised.

4.10 Roster Exception Processing Workflow

4.10.1 Enacted From

This workflow is enacted when there is insufficient correct data to match a roster line item against its corresponding service authorization. This workflow is called from curam.attendance.impl.ProviderRosterLineItemImpl.processRosterLineItem.

4.10.2 Source Location

EJBServer/components/CPM/workflow/ROSTEREXCEPTIONPROCESSING_v1.xml

4.10.3 Default Behavior

This workflow creates a manual activity for a user to review details of a provider roster line item that does not correspond to its associated service authorization. Certain details on a roster line item must match the details on the service authorization associated with the roster line item otherwise any attendance based payments related to the roster will not be paid.

The default implementation of this operation allocates this activity to the roster exception processing work queue for a user to review. Agencies may wish to alter this default behavior, for example, by routing the activity to a different work queue. This workflow creates a manual activity to review the provider roster line item in question. This manual activity is allocated using a function allocation strategy. The modeled operation for this activity is curam.cpm.workflowprocesses.impl.WorkflowAllocationFunction.prliExceptionProcessingAllocationStrategy. The reviewer can choose to make changes to the provider roster line item and submit it for reevaluation or deny/cancel the provider roster line item. Once this activity is completed, the workflow is also completed.

4.10.4 Event Details

The notation of the following event details is as follows:

Table 81. Roster Exception Processing Event Details

Event Raised	Primary Event Data	Raised From
ROSTER.PRLI_PROCESSED	providerRosterLineItemID	curam.attendance.impl.ProviderRosterLineItemImpl.subm
ROSTER.PRLI_CANCELED	providerRosterLineItemID	curam.financial.impl.ServiceInvoiceLineItemImpl.cancel
ROSTER.PRLI_DENIED	providerRosterLineItemID	curam.attendance.impl.ProviderRosterLineItemImpl.deny

4.11 New Client Added to Roster Workflow

4.11.1 Enacted From

This workflow is enacted whenever a provider roster line item is created during creation or modification of a service authorization line item. This workflow is called from see curam.serviceauthorization.impl.ServiceAuthorizationLineItemImpl.generate TaskForNewClientAdded.

4.11.2 Source Location

EJBServer/components/CPM/workflow/NEWCLIENTADDEDTOROSTER_v1.xml

4.11.3 Default Behavior

This workflow creates an activity to send a notification to the owner of a provider roster line item when a client is added to a roster. This notification is sent only during the creation or modification of a service authorization line item which leads to creation of a roster line item.

If the roster line item is submitted or canceled or denied, the corresponding generated notification is removed from the user's task inbox.

4.11.4 Event Details

The notation of the following event details is as follows:

Table 82. New Client Added to Roster Event Details

Event Raised	Primary Event Data	Raised From
ROSTER.PRLI_PROCESSED	providerRosterLineItemID	curam.attendance.impl.ProviderRosterLineItemImpl.submitRoste
ROSTER.PRLI_CANCELED	providerRosterLineItemID	curam.financial.impl.ServiceInvoiceLineItemImpl.cancel
ROSTER.PRLI_DENIED	providerRosterLineItemID	curam.attendance.impl.ProviderRosterLineItemImpl.deny

4.12 Roster Line Item Approval Workflow

4.12.1 Enacted From

This workflow is enacted when a provider roster line item requires manual approval and has reached the "Pending Approval" status. This workflow is called from curam.attendance.impl.ProviderRosterLineItemImpl.approve.

4.12.2 Source Location

EJBServer/components/CPM/workflow/ROSTERLINEITEMAPPROVAL_v1.xml

4.12.3 Default Behavior

This workflow creates a manual activity to review a provider roster line item and approve or deny it. The default implementation of this operation allocates the activity to the roster exception processing work queue. Agencies may wish to alter this default behavior, for example, by routing the activity to a different work queue. This manual activity is allocated using a function allocation strategy. The modeled operation for this is curam.cpm.workflowprocesses.intf.WorkflowAllocationFunction.prliExceptionProcessingAllocationStrategy.

The reviewer can choose to either approve/deny the provider roster line item. Once this activity is completed, the workflow also gets completed.

4.12.4 Event Details

The notation of the following event details is as follows:

Table 83. Roster Line Item Approval Event Wait Activities Details

Event Raised	Primary Event Data	Raised From
ROSTER.PRLI_APPROVED	providerRosterLineItemID	curam.attendance.impl.ProviderRosterLineItemImpl.approve
ROSTER.PRLI_DENIED	providerRosterLineItemID	curam.attendance.impl.ProviderRosterLineItemImpl.deny

4.13 Roster Line Item Correction Approval Workflow

4.13.1 Enacted From

The workflow is enacted whenever a user approves a provider roster line item correction. This workflow is called from curam.attendance.impl.PRLICorrectionImpl.approve.

4.13.2 Source Location

EJBServer/components/CPM/workflow/ROSTERLINEITEMCORRECTIONAPPROVAL_v1.xml

4.13.3 Default Behavior

This workflow contains the processing that is involved in approving a correction made to a provider roster line item. This workflow creates a manual activity to review a provider roster line item correction and approve/deny it. The default implementation of this operation allocates the activity to the roster exception processing work queue.

This manual activity is allocated using a function allocation strategy. The modeled operation for this is curam.cpm.workflowprocesses.intf.WorkflowAllocationFunction.prliExceptionProcessingAllocationStrategy. The reviewer can choose to either approve/deny the provider roster line item correction. Once this activity is completed, the workflow is also completed.

4.13.4 Event Details

The notation of the following event details is as follows:

Table 84. Roster Line Item Correction Approval Event Details

Event Raised	Primary Event Data	Raised From
ROSTER.PRLIC_APPROVED	prliCorrectionID	curam.attendance.impl.PRLICorrectionImpl.approve
ROSTER.PRLIC_DENIED	prliCorrectionID	curam.attendance.impl.PRLICorrectionImpl.deny

Chapter 5. CPM Products and Rule Sets

5.1 Overview

New financial processes have been built for CPM to enable payments to be made to providers. These new processes integrate with existing Cúram Enterprise FrameworkCúram Enterprise Framework (CEF) financial processes. CPM uses Cúram Products and Rule sets for generating the payments for a service provider.

5.2 Products

CPM has following products:

- Provider Invoice

  This product is used to generate the payments for the invoices furnished by the providers.
- Provider Placement

  This product is used to generate the payments related to the placement services offered by the provider.
- Provider Contract

  This product is used to generate the payments that are not dependent on the service utilization.
- Provider Attendance

  This product is used to generate the payments based on the client attendance artifacts provided by the provider for a particular service.

These products are used as a means of getting to Cúram financials rather than them being real benefit products with which a user can interact. All the case processing for these products happens in the back ground on CPM events such as invoice approval, placement of a client, on making a contract live or on provider roster line item approval. We consider the CPM Products has the designated extension point interfaces as the customization points available to an agency.

The DMX files used for the above products are:

- PRODUCT.dmx
- EVIDENCEMETADATA.dmx
- PRODUCTEVIDENCELINK.dmx
- PRODUCTRULESLINK.dmx
- TEMPORALEVIDENCEAPPROVALCHECK.dmx

No changes can be made to these as the generation of financials is dependent upon the product and evidence approval configurations.

5.3 Rule Sets

The list of Rule Sets in CPM is described below. The rule sets can be customized, as long as the customized rule set does not depend on new types of evidence.

Table 85. Payment Type and Rule Set Details

Payment Type	Rule Set Source Location
Provider Invoice	EJBServer\components\CPM\rulesets\Product_51.xml
Provider Placement	EJBServer\components\CPM\rulesets\Product_52.xml

Table 85. Payment Type and Rule Set Details (continued)

Payment Type	Rule Set Source Location
Flat-Rate Contract Payments	EJBServer\components\CPM\rulesets\Product_53.xml
Provider Attendance Payments	EJBServer\components\CPM\rulesets\Product_304.xml

Chapter 6. CPM Financials

6.1 Introduction

CPM financials are developed using the Classic Assessment/reassessment framework, evidence functionality and the Classic Rules Engine.

CPM financials include several tasks such as

- maintenance (creation, approval and activation) of the Product Delivery cases (SILI, Attendance, Placement, Contract) for different types of payments;
- management of evidence using the Evidence functionality;
- execution of Classic Rule Sets;
- assessment/reassessment of financials;
- generation of payments etc.

CPM financials leverage CEF financial processing for assessments and payments.

CPM financial processing is responsible for

- creating and maintaining Evidence for different types of cases;
- creating and maintaining Financial Schedules and transactions associated with different case types;
- processing financial transactions associated with a Participant and/or a Case.

6.2 Payment Types

There are 4 types of Products configured for different payment types in CPM financials

- Service Invoice;
- Placement;
- Flat Rate Contract;
- Attendance

6.2.1 Service Invoice

Service Invoice processing relies on the creation of a Service Authorization when services are allocated to the clients. The individual line items within a Service Authorization can be for a number of different services allocated to that client, which can be provided by different providers. After providing a service, the Provider submits an invoice to the SEM agency. The Provider gets paid once Service Invoice is approved. Service Invoice Line Item payment amounts are treated as evidence for the Service Invoice financial processing.

A product delivery case of type Provider Invoice is created the first time a service invoice line item for a provider is approved. Evidence is created on the case to correspond to the payment amount determined by CPM. The frequency of payment is set based on the established payment frequency for the provider, and leverages CEF functionality around due dates for financial components.

If there is a change in payee, a separate PD case will be created for the payee.

All payments due for the provider for the period will be rolled up and paid as a single payment.

6.2.2 Placement

Placement is a type of service, in which a client is physically placed with the Provider for a period of time. Once a placement service is authorized, a client can be placed with the provider and financials will be started from day one. The unit of measure for the placement will be always a number of days. These placement details will be considered as evidence for processing the placement related financials.

A product delivery case of type Provider Placement is created the first time a placement is made with a provider. The system creates one PD case for each Placement. When a client is transferred within a Provider facility (i.e. form one place to another), this also creates a new product delivery case.

The delivery pattern on the product delivery case is set to a value specified in the property administration section of CPM administration.

For example, if a placement is made for a provider for the first time on June 15th, for the period from June 1st till June 30th, and the frequency is set to the first day of every month, the product delivery case is created on June 15th and the evidence data is set to June 1st till June 30th. The first payment due date is set to July 1st.

6.2.3 Flat rate contract

A Flat Rate Contract is a formal agreement between a Provider and the SEM agency which establishes terms under which services will be delivered. Each contract can cover single or multiple services. All the Contract details are treated as evidence for Flat Rate Contract financial processing.

A product delivery case of type Provider Contract is created the first time a flat rate contract is activated. The system creates one PD case for each Contract per Provider. It also creates a new PD case whenever an existing Contract is renewed.

The information specified in the contract is used to establish a payment schedule for the provider.

6.2.4 Attendance

Attendance rosters are used when services are delivered to the client which require that client attendance be tracked and reported through Attendance Tracking. Attendance is tracked either through a roster submitted by the provider and entered on to the system by an internal user, or by the provider accessing the system externally. Attendance Rosters can be generated automatically based on a configured frequency for a service. Rosters are submitted to the agency after capturing all attendance details. These attendance details are used as evidence for processing the financials.

A product delivery case of type Provider Attendance is created the first time a roster is approved for a provider. Evidence is created on the case to correspond to the payment amount determined by CPM.

The frequency of payment is set based on the established payment frequency for roster based payments. If set, this frequency applies across all providers on the system. If this frequency is not set, the frequency of payment is set based on the established payment frequency for the provider, and leverages CEF functionality around due dates for financial components.

All payments due for the provider for the period will be rolled up and paid as a single payment.

Chapter 7. Service Deliveries

7.1 Introduction

A service delivery is a type of service delivered to a client, which can be created and managed within an integrated case or an outcome plan. These services can be configured to use product delivery processing, Cúram Provider Management (CPM) processing, or a combination of both, depending on how the agency wishes the service to be delivered to the client.

Services which use product delivery processing can use standard product delivery functionality, e.g. eligibility determination for a service and the calculation of payments based on custom rates (a rate which can change over time and can change based on circumstances). Services which use CPM processing can use CPM's financial processing and rate hierarchy. For example, invoices submitted by a provider are matched to a service authorization, and payments are generated based on the provider offering rate, using an out of the box Provider Invoice product delivery case (one per provider). Services which use a combination of both CPM processing and product delivery processing can utilize some or all of the standard features of a product delivery while fully integrating with CPM's service authorization and invoice processing.

If a service offering is configured to use product delivery processing for any aspect of service delivery, a corresponding product must be configured. This chapter outlines the actions and extension points available in CPM to utilize these product delivery features. For general information on configuring a product to be delivered as a service, see Section 3.10 of the `Cúram Integrated Case Management Configuration Guide`. For information on the different delivery types available and the functionality offered by each, see Section 3.8 of the `Cúram Provider Management Guide`.

7.2 Product Design and Configuration

Where service deliveries are configured to use product delivery processing to determine eligibility or payment amounts, the underlying product needs to be associated with a CER rule set and rate tables appropriate to the SEM agency's requirements. For detailed instructions on configuring products and rule sets see the `Cúram How To Build a Product Guide`.

7.3 Rule Set Creation

The rule set used to determine eligibility and calculate payment amounts in respect of the service must be configured to use a combination of client, case, service and invoice or attendance evidence values, depending on the requirements of the agency and the service delivery type. If product delivery processing is being used to determine both eligibility and the payment amount then the recommended approach is to use a separate objective to calculate each of these as follows:-

- The eligibility objective must be configured such that entitlement is determined by checking the value of the relevant attributes, for example, the client's date of birth or employment status. The valueType of the Objective Tag Type for this Objective must be a non money type such as Double to ensure an eligible decision does not result in the generation of financial components, as this is a non-financial objective.
- Entitlement to the payment objective should check for entitlement to the eligibility Objective as well as checking the value of attributes related to custom rates, invoice or attendance evidence. The valueType of the Objective Tag Type for this objective must be Money to ensure the generation of financial components, as this is a financial objective.

7.4 Evidence and Evidence Maintenance

The evidence entities used in the rule set calculations must be configured to use the appropriate propagator type. For example, InvoicePaymentEvidence must be configured to use the `ActiveEvidenceRowRuleObjectPropagator`, ServiceInvoiceLineItem should use the `RuleObjectPropagator`. For detailed instructions on how to configure propagation of different evidence types, see `The Inside Cúram Eligibility and Entitlement Using Cúram Express Rules Guide`.

Evidence types that are used to determine eligibility and calculating payments in respect of services must be configured and associated with the product underlying the service during administration. Shared evidence is maintained at the integrated case level can also be used in rule set calculations.

Changes in evidence values used by the rule set will trigger the assessment engine to run the calculations again resulting in updated decision and payment information.

For detailed information on designing evidence, see the `Cúram Dynamic Evidence Configuration Guide`.

7.5 Custom Rates

If custom rates are to be used to calculate payment amounts in respect of a service, then a rate table must be created and associated with the rule set. For more information on creating and associating rate tables with CER rule sets, see the `Inside Cúram Eligibility and Entitlement Using Cúram Express Rules Guide`.

The value attribute of the Case Decision Objective must be populated in the rule set using values read from your rate tables. Otherwise an appropriate value from CPM such as the amount from an invoice or roster can be used. An attribute to calculate the Estimated Cost can also be included in the rule set where custom rates are used instead of using the default CPM calculation for this value.

7.6 Service Delivery Creation

On creation of a service delivery of type 'Product Delivery', 'Product Delivery with Invoicing' or 'Service Delivery with Eligibility', a product delivery case will be created by the system. This case is an instance of the product type that was configured on the underlying Service Offering. This product delivery is not visible to the user. The caseID of this product delivery is set as the deliveryTypeRelatedID on the service delivery record, and will also be associated with any invoice or attendance payment evidence records associated with the case (i.e., it will be set as the caseID on the associated Evidence Descriptor record). Service deliveries of type Service Delivery will continue to use the caseID of the associated integrated case or outcome plan to populate these fields.

For service deliveries that use product delivery processing to determine eligibility, a hook has been provided to listen for events raised by the Assessment Engine. A default implementation for the `postInsertExamineDecisions` method has been added in `curam.cpm.sl.impl.CPMAssessmentEngineEventListener`, which listens for the creation of new decisions. Where the new decision relates to a service delivery of type 'Product Delivery with Invoicing' or 'Service Delivery with Eligibility' and the decision result is 'Eligible', then a service authorization and any service authorization line items are automatically created for the service delivery. This default behaviour can be altered or enhanced as per agencies own requirements.

7.7 Display of Product Delivery Information

Any product delivery functionality that is related to eligibility and financial processing such as financial transactions, determinations, and evidence is automatically displayed at the service delivery level and can be viewed by a case worker in the context of that service delivery. Other product delivery functionality can also be configured for display if required. For example, certification and appeal details. However,

some development effort is required to display this information. The display of this information must be configured through the use of client navigation files. For more information see Chapter 6 of the Cúram Web Client Reference Manual.

Chapter 8. CPM Taxonomy

8.1 Import Taxonomy File

The process of uploading an external taxonomy file and storing it in the database is called Taxonomy Import Process (TIP). During the process, a single XML file from the database is broken into small, manageable XML chunks. This is done based on the cardinality (one to one, one to many or many to many) of the XML elements defined in that file. These small chunks of XML files (henceforth, called as Cúram Taxonomy files) are created based on criteria like common data shared across various elements.

For example, in AIRS Taxonomy, the Related Concept term elements are shared across Taxonomy term elements. Hence, the TIP creates separate Cúram Taxonomy File for each Related Concept element. Similarly, it also creates separate Cúram Taxonomy Files for each Taxonomy term, Use Reference term (alias for taxonomy term name) etc. These files are stored in the database tables for further processing.

Refer to Appendix B for the structure of Cúram Taxonomy files.

Taxonomy Import Process uses the Java™ -XML binding mechanism to create and publish Cúram Taxonomy Files. The start from Java and XML approach is used for marshaling and un-marshaling XML data from Java to XML and vice-versa, using a mapping XML file. This mapping XML file plays a key role in marshaling/un-marshaling. Published taxonomy is stored in the database. The naming convention for the taxonomy database tables is based on the XML elements defined in the AIRS XML file.

If you want to import your own taxonomy XML file (i.e. other than AIRs taxonomy file) and leverage existing CPM taxonomy subsystem, you should map your XML elements to appropriate elements of the AIRs taxonomy file using Mapping file (refer to Appendix A for TaxonomyMapping.xml) mentioned above. CPM Taxonomy customization can be done in different ways

- Replacing the AIRs XML elements with your XML elements in TaxonomyMapping.xml file
- Replacing your implementation with default implementation.

Note: Currently taxonomy system supports ASCII and UTF-8 encoding, provided encoding format of the database also same

8.1.1 Replacing the AIRs XML elements with your XML elements in TaxonomyTermMapping.xml file

Import process creates the Cúram Taxonomy files using TaxonomyMapping.xml file. Cúram Taxonomy files (Refer to the Appendix B for the structure of these XML files) are as follows:

- TaxonomyTerm.xml
- RelatedConcept.xml
- UseReference.xml
- ExternalTerm.xml

Currently TaxonomyMapping.xml contains the mapping information between AIRs taxonomy elements and Cúram Taxonomy files elements along with respective bonded java classes. If you want leverage the existing taxonomy functionality, you should map your elements to the respective elements defined in the TaxonomyMapping.xml file. Along with this mapping, you should also override the APPRESOURCE.dmx with your TaxonomyMapping.xml file location.

Following table provides the mapping of AIRs taxonomy elements with the respective Cúram Taxonomy files elements and their associated java classes in TaxonomyMapping.xml.

Table 86. AIRs and Curam Taxonomy Mapping Elements.

This table describes mapping between AIRs Taxonomy Elements and Curam Taxonomy Elements.

AIRS Taxonomy Element	Description	Cúram Taxonomy Element	Cúram Taxonomy File	Java Representation of the Element
taxonomy	Root element of taxonomy term	record	TaxonomyTerm.xml	curam.taxonomy.util.impl.Taxonomy
name	Name of the taxonomy term	name	TaxonomyTerm.xml	java.lang.String
definition	Definition of the taxonomy term	definition	TaxonomyTerm.xml	java.lang.String
facet	Facet of the taxonomy term	facet	TaxonomyTerm.xml	java.lang.String
comments	Detailed description of the taxonomy term	comments	TaxonomyTerm.xml	java.lang.String
bibliographicReference	Bibliographic Reference of the taxonomy element	bibliographicReference	TaxonomyTerm.xml	java.lang.String
createdDate	Creation date of the taxonomy term	createdDate	TaxonomyTerm.xml	java.lang.String
lastModifiedDate	Last modified date of the taxonomy term	lastModifiedDate	TaxonomyTerm.xml	java.lang.String
externalTerms	External System Classification terms associated with the taxonomy term	externalTerms	ExternalTerm.xml	curam.taxonomy.util.impl.ExternalTerm
relatedConcepts	Related Concepts associated with the taxonomy term	relatedConcepts	RelatedConcept.xml	curam.taxonomy.util.impl.RelatedConcep
useReferences	Alias names of the taxonomy term	useReferences	UseReference.xml	Alias names of the taxonomy term
relatedTerms	Related Terms of the taxonomy term	relatedTerms	TaxonomyTerm.xml	java.lang.String
oldCodes	Old codes associated with the taxonomy term	oldCodes	TaxonomyTerm.xml	java.lang.String

Update the TaxonomyMapping.xml by replacing AIRS Taxonomy Elements with elements your Taxonomy xml file for creating Cúram Taxonomy files. These files will be used by Taxonomy-in-edit process to make changes to the content of xml files and Taxonomy Publish process to persist xml content to the relational tables.

Taxonomy Publish process uses the same mappings file for un-marshalling the XML content to java objects before persisting to the database. It persist the content of Cúram Taxonomy files to respective tables. Following table explains how content of different Cúram Taxonomy files is spawned across different tables:

Table 87. Mapping between Curam Taxonomy File Data and Curam Relational Entities.

This table describes mapping between Curam Taxonomy files and Curam Relational Entities.

Cúram Taxonomy file	Entities
TaxonomyTerm.xml	TaxonomyTerm,TaxonomyTermNameLink,LocalizableText,TextTranslation,
ExternalTerm.xml	ExtSysClassification,ExtSysClassifnTermLink,LocalizableText,TextTranslati
RelatedConcept.xml	RelatedConcept,RelatedConceptNameLink,LocalizableText,TextTranslation
UseReference.xml	UseReference,LocalizableText,TextTranslation,UseReferenceNameLink

8.1.2 Replacing your implementation with default implementation

If you want to add your own taxonomy elements other than elements defined in TaxonomyMapping.xml, you should follow the Replaceable Implementation approach. Current taxonomy system doesn't support hooking of partial implementation for processing your extra element. Since taxonomy system is developed based on Design by Contract principle, you can replace default implementation with custom implementation without violating the contract (i.e. strictly constrained to the interface APIs). Custom implementation is only required if the structure of the following Cúram Taxonomy files are changed, that is, adding new elements or deleting existing elements

TaxonomyTerm.xml,RelatedConcept.xml,UseReference.xml,ExternalTerm.xml

You have to provide custom implementation to the following interfaces

- curam.taxonomy.impl.ImportTaxonomy
- curam.taxonomy.impl.PublishTaxonomy
- curam.taxonomy.impl.TaxonomyInEditData
- curam.taxonomy.impl.TaxonomyInEditDataConverter

8.2 Retrieve Indexes and Retrieve Taxonomy Terms By Keywords

Taxonomy Search functionality provides a number of service layer interfaces that are specifically designed for customization. A new custom implementation can be provided for any of the interfaces curam.taxonomy.sl.search.impl. TaxonomySearch. It is worth noting that default implementations which uses the Cúram Generic Search Server are provided for these interfaces. Please read chapter 2 (Using Strategy Patterns to Customize CPM) for more details

Chapter 9. Compliancy

9.1 CPM Sample

CPM contains a sample component that is primarily added with following purposes:

- to help the testing team test CPM APIs
- to help the development team test CPM extension points.

Since this component is added only with above purposes, use or customization of this component is not supported.

CPM Sample component has 3 packages which should not be used or customized.

- curam.cpmsample.changecases.impl:- This package is mainly used for testing of extension points in CPM.
- curam.cpmsample.facade.impl:- This package has façade classes and the associated client directory is components/CPMSample
- curam.cpmsample.impl:- This has Module class. It is again used for testing of extension points in CPM.

9.2 Miscellaneous Entities

CPM created following new entities in CPM component to add a new feature which supports multiple clients for provider roster line item. As this feature is not supported by the application currently, these entities may change in the future. It is highly recommended that these entities are not used.

- PRLIClient
- PRLIClientHistory
- PRLICorrectionClient

Appendix. Appendix

A.1 Appendix A

The structure of the xml is based on the Castor v0.9.5.4 Mapping xml schema.

```
<?xml version="1.0" encoding="UTF-8"?>
            <mapping>
                <class auto-complete="false"
                name="curam.taxonomy.util.impl.Taxonomy">
                    <map-to xml="taxonomy" />
                    <field collection="arraylist" name="taxonomyTerms"
                    type="curam.taxonomy.util.impl.TaxonomyTerm">
                        <bind-xml name="record" />
                    </field>
                </class>
                <class auto-complete="false"
                name="curam.taxonomy.util.impl.TaxonomyTerm">
                    <map-to xml="record" />
                    <field name="name" type="java.lang.String">
                        <bind-xml name="name" node="element" />
                    </field>
                    <field name="code" type="java.lang.String">
                        <bind-xml name="code" node="attribute" />
                    </field>
                    <field name="definition" type="java.lang.String">
                        <bind-xml name="definition" node="element" />
                    </field>
                    <field name="facet" type="java.lang.String">
                        <bind-xml name="facet" node="element" />
                    </field>
                    <field name="comments" type="java.lang.String">
                        <bind-xml name="comments" node="element" />
                    </field>
                    <field name="bibliographicReference" type="java.lang.String">
                        <bind-xml name="bibliographicReference" node="element" />
                    </field>
                    <field name="createdDate" type="java.lang.String">
                        <bind-xml name="createdDate" node="element" />
                    </field>
                    <field name="lastModifiedDate" type="java.lang.String">
                        <bind-xml name="lastModifiedDate" node="element" />
                    </field>
                    <field collection="arraylist" name="taxonomyTerms"
                    type="curam.taxonomy.util.impl.TaxonomyTerm">
                        <bind-xml name="record" />
                    </field>
                    <field collection="arraylist" name="externalTerms"
                    type="curam.taxonomy.util.impl.ExternalTerm">
                        <bind-xml name="externalTerm" />
                    </field>
                    <field collection="arraylist" name="relatedConcepts"
                    type="curam.taxonomy.util.impl.RelatedConcept">
                        <bind-xml name="relatedConcept" />
                    </field>
                    <field collection="arraylist" name="useReferences"
                    type="java.lang.String">
                        <bind-xml name="useReference" />
                    </field>
                    <field collection="arraylist" name="relatedTerms"
                    type="java.lang.String">
                        <bind-xml name="see Also" />
                    </field>
                    <field collection="arraylist" name="oldCodes"
                    type="java.lang.String">
                        <bind-xml name="oldCode" />
                    </field>
                </class>
                <class auto-complete="false"
                name="curam.taxonomy.util.impl.RelatedConcept">
                    <map-to xml="relatedConcept" />
                    <field name="code" type="java.lang.String">
                        <bind-xml name="code" node="attribute" />
                    </field>
                    <field name="name" type="java.lang.String">
                        <bind-xml node="text" />
                    </field>
                </class>
                <class auto-complete="false"
```

A.2 Appendix B: Schema definitions of the XML fragments created by Import process.

A.2.1 TaxonomyTerm.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
                <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
                elementFormDefault="qualified">
                    <xs:element name="record">
                        <xs:complexType>
                            <xs:sequence>
                                <xs:element ref="name" />
                                <xs:element ref="definition" />
                                <xs:element ref="facet" />
                                <xs:element ref="comments" />
                                <xs:element ref="bibliographicReference" />
                                <xs:element ref="createdDate" />
                                <xs:element ref="lastModifiedDate" />
                                <xs:element ref="relatedTerms" />
                                <xs:element ref="oldCode" />
                            </xs:sequence>
                            <xs:attribute name="code" use="required"
                            type="xs:string" />
                        </xs:complexType>
                    </xs:element>
                    <xs:element name="name">
                        <xs:complexType mixed="true">
                            <xs:attribute name="locale" use="required"
                            type="xs:string" />
                        </xs:complexType>
                    </xs:element>
                    <xs:element name="definition">
                        <xs:complexType mixed="true">
                            <xs:attribute name="locale" use="required"
                            type="xs:string" />
                        </xs:complexType>
                    </xs:element>
                    <xs:element name="facet" type="xs:string" />
                    <xs:element name="comments">
                        <xs:complexType>
                            <xs:attribute name="locale" use="required"
                            type="xs:string" />
                        </xs:complexType>
                    </xs:element>
                    <xs:element name="bibliographicReference">
                        <xs:complexType>
                            <xs:attribute name="locale" use="required"
                            type="xs:string" />
                        </xs:complexType>
                    </xs:element>
                    <xs:element name="relatedTerms">
                        <xs:complexType>
                            <xs:attribute name="locale" use="required"
                            type="xs:string" />
                        </xs:complexType>
                    </xs:element>
                    <xs:element name=" oldCode " type="xs:string" />
                    <xs:element name="createdDate" type="xs:string" />
                    <xs:element name="lastModifiedDate" type="xs:string" />
                </xs:schema>
```

A.2.2 UseReference.xsd

```
<?xml version="1.0" encoding="UTF-8"?>

                <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
                elementFormDefault="qualified">
                    <xs:element name="useReferences">
                        <xs:complexType>
                            <xs:sequence>
                                <xs:element ref="useReference" />
                            </xs:sequence>
                        </xs:complexType>
                    </xs:element>
                    <xs:element name="useReference">
                        <xs:complexType>
                            <xs:sequence>
                                <xs:element ref="text" />
                            </xs:sequence>
                        </xs:complexType>
                    </xs:element>
                    <xs:element name="text">
                        <xs:complexType>
                            <xs:simple Content>
                                <xs:extension base="xs:string">
                                    <xs:attribute name="locale" use="required"
                                    type="xs:string" />
                                </xs:extension>
                            </xs:simple Content>
                        </xs:complexType>
                    </xs:element>
                </xs:schema>
```

A.2.3 RelatedConcept.xsd

```
<?xml version="1.0" encoding="UTF-8"?>

                <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
                elementFormDefault="qualified">
                    <xs:element name="relatedConcept">
                        <xs:complexType>
                            <xs:sequence>
                                <xs:element ref="name" />
                            </xs:sequence>
                            <xs:attribute name="code" use="required"
                            type="xs:NCName" />
                        </xs:complexType>
                    </xs:element>
                    <xs:element name="name">
                        <xs:complexType mixed="true">
                            <xs:attribute name="locale" use="required"
                            type="xs:string" />
                        </xs:complexType>
                    </xs:element>
                </xs:schema>
```

A.2.4 ExternalTerm.xsd

```
<?xml version="1.0" encoding="UTF-8"?>

                <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
                elementFormDefault="qualified">
                    <xs:element name="externalTerm">
                        <xs:complexType>
                            <xs:sequence>
                                <xs:element ref="name" />
                                <xs:element ref="system" />
                            </xs:sequence>
                            <xs:attribute name="code" use="required"
                            type="xs:string" />
                        </xs:complexType>
                    </xs:element>
                    <xs:element name="name">
                        <xs:complexType mixed="true">
                            <xs:attribute name="locale" use="required"
                            type="xs:string" />
                        </xs:complexType>
                    </xs:element>
                    <xs:element name="system" type="xs:string" />
                </xs:schema>
```

Notices

This information was developed for products and services offered in the U.S.A. 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

Armonk, NY 10504-1785

U.S.A.

For license inquiries regarding double-byte (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

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: 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 states 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 Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply 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 Corporation

Dept F6, Bldg 1

294 Route 100

Somers NY 10589-3216

U.S.A.

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.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

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-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All 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 the names and addresses used by an actual business enterprise 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.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

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/us/en/copytrade.shtml.

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

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

Printed in USA