



Cúram 8.1.2

Common Intake Configuration Guide

Note

Before using this information and the product it supports, read the information in [Notices on page 53](#)

Edition

This edition applies to Cúram 8.1, 8.1.1, and 8.1.2.

© Merative US L.P. 2012, 2024

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

Contents

Note.....	iii
Edition.....	v
1 Configuring Common Intake.....	9
1.1 Configuring the intake process.....	9
Configuring programs for an intake process.....	10
Configuring initial contact with the individual.....	11
Configuring the cases view for an individual.....	12
Configuring the case quick search.....	14
1.2 Customizing the intake application workflow.....	16
Customizing the generic PDF summary form for processed applications.....	19
1.3 Defining triage.....	22
1.4 Defining screenings.....	23
Setting up a Screening Definition.....	23
Adding urgent alerts to the rule set.....	37
Recommending programs.....	37
1.5 Configuring client applications.....	38
Configuring the programs covered by an application type.....	39
Integrating the application with Universal Access.....	41
Configuring workflows for application processing.....	42
Configuring the texts associated with an application.....	43
Configuring cases to which the application can be transferred.....	44
Configuring the cases view for the application.....	45
Configuring the cases to which the application can be added.....	46
Configuring the case status after the application has been transferred.....	47
Configuring the case status after undoing application transfer.....	48
Configuring the application TODO list context.....	49
Configuring the application quick search.....	49
1.6 Common intake application properties.....	50
Notices.....	53
Privacy policy.....	54
Trademarks.....	54

1 Configuring Common Intake

Configure the intake process with an intake process definition XML document. Intake processes can be assigned to user roles. Programs and services can be defined for an intake. A triage definition can be created.

Common Intake (CCI) allows workers in an agency to perform the functions that are required as part of the intake process. CCI also provides a collaborative, 'No Wrong Door' approach to the intake process for clients who applying for benefits. This process reduces the number of contacts a client makes to access benefits across multiple agencies, by providing workers with the facility to perform the intake process across multiple programs.

1.1 Configuring the intake process

Intake processes differ by agency. In particular, the intake process for needs-based programs such as cash assistance and unemployment benefits differs from the intake process for a protection-based program such as child welfare.

Introduction

For needs-based programs, the client usually completes an application form, and the agency determines eligibility for programs based on the information that is provided in the application. Common Intake provides an intake process that can be configured to suit both protection and needs based programs.

For protection-based programs, the process is triggered by a report of abuse or neglect. The agency then investigates the report, and puts measures in place to ensure that the client is safe and is not at risk of further abuse or neglect.

The intake process in Common Intake is represented by an XML document, which is stored in the database. This XML document controls certain aspects of the behavior of the application including certain flow of control aspects. Different users, or rather the applications that are assigned to those users, might be assigned different intake process XML documents in configuration, allowing for varying behavior between those users.

Intake process definition

The intake process is defined in two database tables. The *IntakeProcessName* database table specifies a named intake process definition. This is a logical name that can be used to uniquely reference the intake process. Associated with the *IntakeProcessName* database table is an entry in the *IntakeProcessConfiguration* database table. This database table contains the actual XML document that defines this particular intake process.

Assigning an intake process to a user

The intake process that is defined in the *IntakeProcessName* and *IntakeProcessConfiguration* database tables can be assigned to user roles. This is achieved by creating an entry in the *IntakeProcessRoleAssignment* database table and associating the intake process definition with an application code. As each user has one and only one application code that is associated with them, this creates an association between an intake process and all users with that application code.

System owned transactions, such as the deferred process instigated when an application is submitted, require access to a default intake process configuration. To ensure that a configuration is available when the system needs to access one, add an entry to the *IntakeProcessRoleAssignment* database table assigning a default intake process to the 'DefaultApp' application code.

Creating the intake process XML document

For the Intake Process to have any behavior, you must write an intake process configuration XML document, this is the XML document that is loaded into the *IntakeProcessConfiguration* database table. To create this XML document:

- Create an XML document and save it to a location on disk.
- The root tag of the XML document is 'IntakeProcess'. Create an element with this name as the root element of the XML document.

Configuring programs for an intake process

Configure the intake process with a set of programs that are recognized for the intake. These programs are the only programs that are supported by the intake process and must be explicitly declared in the XML document.

Configuring the intake process with a set of programs

To specify programs supported by the intake process, take the following steps:

- Open the XML document that represents the intake process configuration.
- Create or locate an XML element *GlobalConfiguration* as a direct child of the *IntakeProcess* element, that is, as a direct child of the root element of the XML document. Only one global configuration element can exist in the document.
- For each program that the intake process supports, create an element *ProgramType* as a direct child of the *GlobalConfiguration* element with the following attributes:

Table 1: ProgramType Attributes

This table describes the *ProgramType* element attributes for an intake.

Attribute Name	Type	Purpose
<i>ProgramType</i>	<i>CodeTableCode</i>	This attribute contains an entry from the code table <i>ProgramType</i> . This code table specifies a program type, for example, medical assistance. This attribute is mandatory and must be specified.
<i>ProgramTypeID</i>	<i>Long</i>	This attribute is used to integrate Common Intake with Merative™ Cúram Universal Access. This attribute contains the primary key of the database table <i>ProgramType</i> , which is used by Universal Access to specify the programs that it manages. This attribute is only specified if <i>ProgramType</i> is also used for Universal Access, and only if integration with Universal Access is required.

Configuring the description for a program

Each program that is specified must have a display description. This is a short description of the programs purpose that is displayed to the user when viewing the program for selection on list

pages in the application. This text is configured in a property file that is loaded in to the resource store. To configure the descriptions:

- Create a property file resource in the resource store with the following name `<IntakeProcessName>GlobalConfiguration`. Replace `<IntakeProcessName>` with the logical name of the Intake Process as stored in the `IntakeProcessName` database table.
- For each program that requires a description, create a property in the property file with the following name: `<ProgramTypeCode>.Description` and specify the description text as the value of the property. Replace `<ProgramTypeCode>` with the code from the `ProgramType` code table that represents the Program.

Configuring initial contact with the individual

Initial contact occurs when a person walks into the organization's office to discuss their needs. The worker then searches for the person and locates their participant record (if one exists), or registers them.

When the person record is found, the worker is presented with the individual's home page, which presents a summary of the person's intake so far. They are also presented with several tabs for the person, allowing them to perform several functions such as recording a triage, performing a screening, recording a life event and so on.

Configuring the intake summary

The intake summary provides a view of the users history relevant to Common Intake. The **Intake Summary** page is generally not configurable. However, it does have one configurable aspect. It is possible to specify which clusters are displayed for the intake configuration as follows:

- Open the XML document that represents the intake process configuration.
- Create or locate an XML element `IndividualConfiguration` as a direct child of the `IntakeProcess` element, that is, as a direct child of the root element of the XML document. Only one `IndividualConfiguration` element might exist in the document.
- Create or locate an XML element `IntakeSummary` as a direct child of the `IndividualConfiguration` element previously located.

On this element, the following attributes can be specified which configure the behavior of the **Intake Summary** page.

Table 2: Intake Summary Attributes

This table describes configuration attributes for the **Intake Summary** page.

Attribute Name	Type	Purpose
<code>ShowPendingApplicationCluster</code>	Boolean	Controls whether the list of pending applications are displayed on the home page. If set to true the cluster is displayed.
<code>ShowCurrentCluster</code>	Boolean	Controls whether the list of current cases that are associated with the Individual is displayed on the home page. If set to true the cluster is displayed.

Note: You can use standard customization techniques to replace the Common Intake version of this page with a custom version. This can be achieved by using the standard approach for customizing UIM pages. There are two consequences to this approach:

- As the approach overrides the version that is shipped by Cúram, the configurable settings do not necessarily take effect on the custom page that is defined, unless the same approach to the page development is taken.
- Most importantly, future updates to the page made by Cúram do not automatically appear in the customized page. As a result future improvements to this page and the configurability of this page is not immediately available.

Configuring the cases view for an individual

As part of the intake process for an individual a **Cases** tab is available that allows any cases that are associated with the client to be viewed.

Configuring the Cases view

Certain information on the **Cases** tab is configurable and these configurations are described. To configure the cases, view for an individual do the following:

- Open the XML document that represents the intake process configuration.
- Create or locate an XML element *IndividualConfiguration* as a direct child of the *IntakeProcess* element, that is, as a direct child of the root element of the XML document. Only one *IndividualConfiguration* element can exist in the document.
- Create or locate an XML element *ShowExistingCase* as a direct child of the *IndividualConfiguration* element previously located.

Configuring case preview panels

Preview panels for the cases that are listed in the **Cases** tab can be configured based on case type as follows:

- For each preview panel you want to configure on the individual cases list, create an element *PreviewPanel* as a direct child of *ShowExistingCase*.
- On this element, the following attributes can be specified which configures the page to display on the individual cases list preview panel for the defined case type:

Table 3: *PreviewPanel* Attributes

This table describes configuration attributes for the individual tab case preview panels.

Attribute Name	Type	Purpose
caseType	Code Table Entry	This attribute contains an entry from the case type code table.
Page	String	This attribute contains the name of the client page that is used for the preview panel.

Configuring case types

The types of cases that are displayed on the list can be configured as follows:

- For each case type you want to display on the individual cases list, create an element *ShowCasesOfType* as a direct child of *ShowExistingCase*.
- On this element, the following attribute can be specified which configures the type of case to be displayed on the individual cases list:

Table 4: *ShowCasesOfType* Attributes

This table describes configuration attributes for the types of cases on the individual tab cases list.

Attribute Name	Type	Purpose
Type	Code Table Entry	Contains an entry from the case type code table.

Configuring case statuses

Configure case statuses that are displayed on the **Cases** tab as follows:

- For each case status you want to display on the individual cases list, create an element *ShowCasesOfStatus* as a direct child of *ShowExistingCase*.
- On this element, the following attribute can be specified which configures the status of case to be displayed on the individual cases list:

Table 5: *ShowCasesOfStatus* Attributes

This table describes configuration attributes for the statuses of cases on the individual tab cases list.

Attribute Name	Type	Purpose
Status	Code Table Entry	This attribute contains an entry from the case status code table.

Configuring undisposed applications

Configure whether cases with undisposed applications are displayed as follows:

- Create an element *ShowCasesWithUndisposedApplication* as a direct child of *ShowExistingCase*.
- On this element, the following attribute can be specified which configures whether cases with undisposed applications are displayed on the individual cases list.

Table 6: *ShowCasesWithUndisposedApplications* Attributes

This table describes configuration attributes defining whether cases with undisposed applications are displayed.

Attribute Name	Type	Purpose
Value	Boolean	Controls whether cases with undisposed applications are displayed on the individual cases list.

Configuring the case quick search

Configure the results of the quick search for cases for the intake process.

Configuring case quick search results

Configure the case quick search results as follows:

- Open the XML document that represents the intake process configuration.
- Create or locate the XML element *ExistingCase* as a direct child of the *IntakeProcess* element.
- Create or locate an XML element *QuickSearch* as a direct child of the *ExistingCase* element.

Configuring case quick search preview panels

Configure preview panels for the cases in the results list based on case type as follows:

- For each preview panel that you want to configure on the search create an element *PreviewPanel* as a direct child of *QuickSearch*.
- Specify the following attributes on this element that configure the page to display in the **quick search results** preview panel for the defined case type:

Table 7: *PreviewPanel* Attributes

Attribute Name	Type	Purpose
<i>caseType</i>	Code Table Entry	Contains an entry from the case type code table.
<i>Page</i>	String	Contains the name of the client page that is used for the preview panel.

Configuring case types

Configure the types of cases that are displayed on the **case quick search results** panel as follows:

- For each case type, you want to display in the quick search results create an element *ShowCasesOfType* as a direct child of *QuickSearch*.
- Specify the following attribute on this element, which configures the type of case to be displayed in the quick search results:

Table 8: *ShowCasesOfType* Attributes

Attribute Name	Type	Purpose
<i>Type</i>	Code Table Entry	Contains an entry from the case type code table.

Configuring case statuses

Configure the statuses of cases that are displayed on the case quick search results as follows:

- For each case status, you want to display in the quick search results create an element *ShowCasesOfStatus* as a direct child of *QuickSearch*.
- Specify the following attribute on this element the following attribute, which that configures the status of case to be displayed in the quick search results:

Table 9: ShowCasesOfStatus Attributes

Attribute Name	Status	Purpose
<i>Type</i>	Code Table Entry	Contains an entry from the case status code table.

Configuring undisposed applications

Configure whether cases with undisposed applications are displayed as follows:

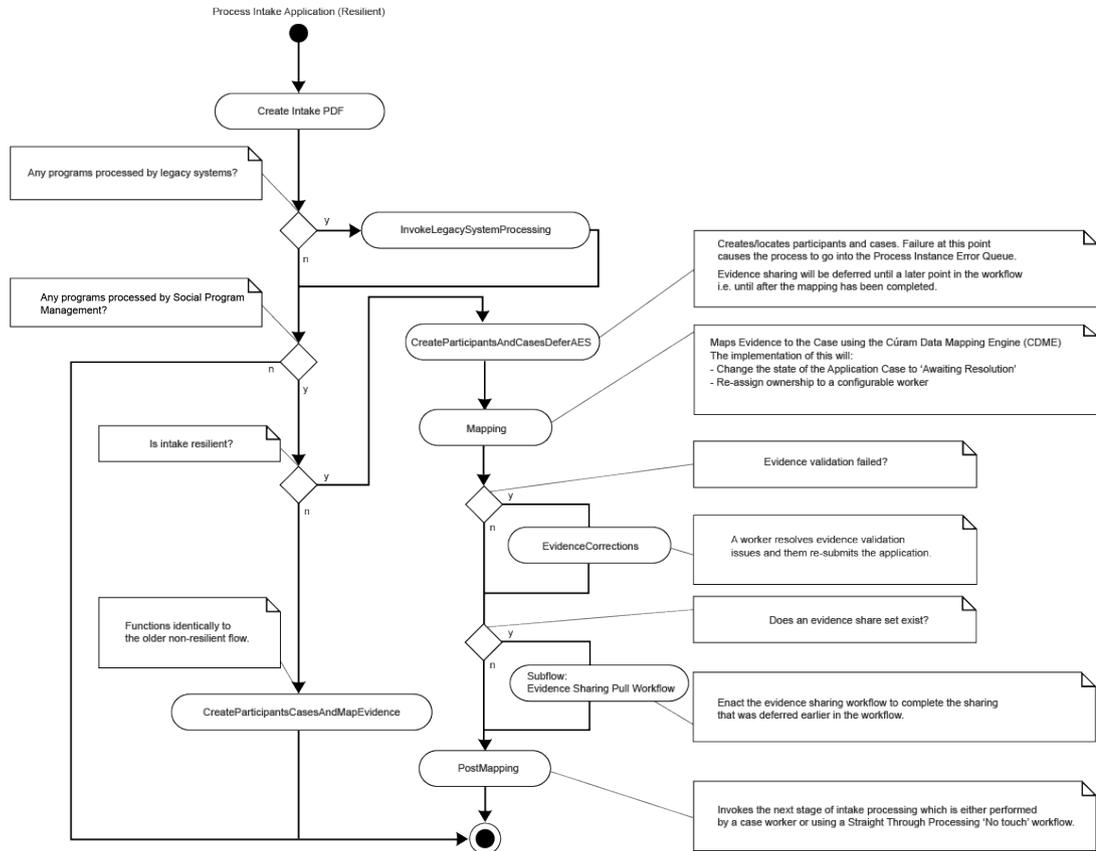
- Create an element *ShowCasesWithUndisposedApplication* as a direct child of QuickSearch.
- Specify the following attribute on this element, which configures whether cases with undisposed applications are displayed on the cases quick search results.

Table 10: ShowCasesWithUndisposedApplications Attributes

Attribute Name	Type	Purpose
<i>Value</i>	Boolean	Controls whether cases with undisposed applications are displayed on the cases quick search results list.

1.2 Customizing the intake application workflow

Review a summary of the intake application workflow in a flowchart.



- **Create intake PDF**
This automatic activity creates a PDF document based on the content of the application. For more information, see [Customizing the generic PDF summary form for processed applications on page 19](#).
- **InvokeLegacySystemProcessing**
This automatic activity sends applications to legacy systems by using web services. This path is taken only if there are legacy systems that are associated with at least one of the programs on the application.
- **CreateParticipantsAndCasesDeferAES**
This automatic activity creates participants for the submitted application and then creates one or more cases for the programs that are associated with the application. In most cases, application cases are created.
This path is taken only if the configuration property `curam.intake.use.resilience` is set to `true`. For compatibility with previous versions, this property is `false` by default. However, it is recommended that all production systems set this value to `true`. For more information about the implications of setting this value to `true`, see .

This activity also defers the evidence sharing that occurs when case members are added to an application case to a later point in the workflow.

Note: For customizations to this workflow or for custom intake workflows, evidence sharing can be deferred by using the `curam.aes.sl.observe.impl.AESShareSetPullManualEnactment.process()` API. For more information, see the associated Javadoc.

- **Mapping**
This automatic activity uses the Cúram Data Mapping Engine (CDME) to map data collected in the application script into Case Evidence. If a validation issue occurs with the mapped evidence, this activity is automatically retried. During the retry, if there is a single Application Case, the validations are disabled and a WDO flag `IntakeCaseDetails.mappingValidInd` set to `false`.
- **EvidenceCorrections**
This manual task is called if the Mapping activity fails due to a validation error. That is, the `IntakeCaseDetails.mappingValidInd` set to `false`. The assignment of this task is configurable.

For more information about the ownership strategy, see the *Intake Developer's Guide*. The caseworker or operator must resolve the evidence validation issues and resubmit the application.
- **Subflow: Evidence Sharing Pull Workflow**
If there is evidence to share, the evidence sharing pull workflow is enacted to trigger any sharing activities that were deferred in `CreateParticipantsAndCasesDeferAES`. If there is no evidence to share, the workflow bypasses this step.
- **PostMapping**
This automatic activity starts the next stage of application processing by calling the `IntakeApplication.IntakeApplicationEvents.postMapDataToCuram()` event.
- **CreateParticipantsCasesAndMapEvidence**
This path is followed when the configuration property `curam.intake.use.resilience` is set to `false`. This automatic activity behaves identically to the older non-resilient workflow. It creates cases and participants and completes all evidence mapping in a single transaction, which results in a less resilient process if a failure occurs.

Customers can customize the workflow compliantly as described in the *Universal Access Responsive Web Application Guide* and the *Web Client Reference Manual*. Customers must not change the enactment structs that are used by these workflows.

Customizing the generic PDF summary form for processed applications

By default, the PDF summary is automatically generated from an XSL template for all intake applications.

The PDFs are rendered by the XML server. For more information about XSL templates, see the *Communications Overview Guide*.

Configuring the generic PDF summary template

Complete the following steps to change or update the generic PDF summary template for intake applications. If needed you can upload your own custom template. Customers can choose between an improved template or the default earlier template.

About this task

- ***WSXSLTEMPLATEINST001.xsl***

The `/EJBServer/components/WorkspaceServices/data/initial/blob/WSXSLTEMPLATEINST001.xsl` template is configured by default and provides compatibility for existing scripts.

- ***IntakeApplicationPDFTemplate.xsl***

Customers can use the more recent `/EJBServer/components/WorkspaceServices/data/initial/blob/IntakeApplicationPDFTemplate.xsl` template to generate an improved summary that is based on the final summary page of the IEG application script.

The PDF summary template does not support some IEG scripts. To define a list of scripts to be excluded from summary PDF template generation, if a summary PDF template is configured, use the `curam.intake.pdftemplate.iegsummarypagelayout.scriptexcludelist` environment variable. For the excluded IEG scripts, the default data store PDF template is used instead to generate PDF documents.

The PDF summary template does not support the following IEG scripts:

- Scripts with no summary pages.
- Scripts with one or more summary pages, but where a summary page isn't at the end of the script.
- Scripts with a summary page that is nested in a condition.
- Scripts where the last summary page contains no cluster.

For a summary PDF to be generated from an IEG script by using the `IntakeApplicationPDFTemplate.xsl` template, a summary page must be at the end of the script. If your IEG scripts don't have a summary page, you must either add a summary page or use the default `WSXSLTEMPLATEINST001.xsl` template, which doesn't need a summary page in the IEG script.

You can configure the text and logo in the header, and further customize the template to define the information that you want to include in the PDF summary form.

Procedure

1. Log in as a system administrator, go to **Communications > XSL Templates**.
2. Search for the **Intake Application** template with the following details:
 - Description: Intake Application
 - Relates To: Intake Application
 - Category: Intake Application
 - Subcategory: Intake Application
 - Template Type: Letter
 - Template ID: Intake Application
3. Select the template and check it out.
The existing template version is saved in case you need it later.
4. Locate the template that you want to upload.
This can be your own custom template, or the improved template. For example, to use the improved *IntakeApplicationPDFTemplate.xsl* template, download it from `/EJBServer/components/WorkspaceServices/data/initial/blob/IntakeApplicationPDFTemplate.xsl` so you can upload it here.
5. Click **Check in template**, and upload the new or updated template.

Customizing generic PDF summary forms based on the *IntakeApplicationPDFTemplate.xsl* template

You can customize the generic PDF summary form of an online application that is based on the most recent *IntakeApplicationPDFTemplate.xsl* template. You can customize the text and logo in the header, and customize the XSL template to define the information that you want to include in the generic PDF summary form.

Before you begin

The *IntakeApplicationPDFTemplate.xsl* template uses information from the summary page in IEG scripts to generate the PDF summary form, see the *Universal Access Configuration Guide*.

Where an IEG script contains multiple conditional summary pages, the last visible summary page is used to generate the PDF. If your IEG scripts don't have a summary page, you must either add a summary page, or use the earlier *WSXSLTEMPLATEINST001.xsl* template, which doesn't need a summary page.

Procedure

1. To customize the PDF header, edit the `intake-application-pdf-props` app resource, which is located in the `EJBServer/components/WorkspaceServices/data/initial/blob/prop/IntakeApplicationPDFProps.properties` file.

You can customize the following information:

```
# START NON-TRANSLATABLE
pdf.logo=logo-pdf
# END NON-TRANSLATABLE
pdf.logo.altText=logo
pdf.application.reference=Application number:
pdf.submitted.date=Submitted on
```

- a) Update the text as needed.

- b) To customize the PDF logo, add an image resource that matches the name on the `pdf.logo` property and locale. The default name of the image resource is `logo-pdf`.
2. To update the template, search for an XSL resource that is called `Intake Application`, which is located at:

```
EJBServer/components/WorkspaceServices/data/initial/blob/
IntakeApplicationPDFTemplate.xsl
```

This XSL uses XSL-FO 1.0 technology to generate the PDF. For more information about XSL, see <https://www.w3.org/TR/xsl11>.

3. Update the system to use your custom template, see [Configuring the generic PDF summary template on page 19](#).

Customizing generic PDF summary forms based on the `WSXSLTEMPLATEINST001.xsl` template

Complete the following step to customize a PDF summary form that is based on the default `WSXSLTEMPLATEINST001.xsl` template. You can customize the XSL template to define the information that you want to include in the generic PDF summary form.

About this task

The data passed to the XSL template reads from the data store. For more information, see the *Communications Overview Guide*.

Procedure

1. Instead of displaying the data store labels in the PDF, you can define a property file to specify alternative names for entities and attributes and to hide entities and attributes that you do not want to display in the PDF.

Define a property file with the naming convention `<application schema name>PDFProps`, and edit the contents of the property file as follows:

- **Name an entity**
`<Entity Name=<Name To Be Displayed in the PDF>, for example, Application=Intake Application`
- **Hide an entity**
`<Entity Name.hidden=true, for example, ScreeningType.hidden=true`
- **Hide an attribute**
`<Entity Name.Attribute Name.hidden=true, for example, Application.userName.hidden=true`
- **Specify a label for an attribute**
`<Entity Name.Attribute Name=PDF Label, for example, Submission.digFirstName=First Name`

Upload the property file to **Application Resources** in the **Intelligent Evidence Gathering** section of the administration application. For more information about IEG, see the *Working with Intelligent Evidence Gathering (IEG) Guide*.

2. To update the template, search for an XSL resource that is called `Intake Application`, which is located at:

```
/EJBServer/components/WorkspaceServices/data/initial/blob/WSXSLTEMPLATEINST001.xsl
```

Update the template as needed. This XSL uses XSL-FO 1.0 technology to generate the PDF. For more information about XSL, see <https://www.w3.org/TR/xsl11>.

1.3 Defining triage

Triage allows a caseworker to quickly gauge a client's needs by asking a core set of fundamental questions. The result allows the caseworker to understand the client's situation in relation to the core needs. A triage can be performed by the caseworker resulting in a list of suitable services, service providers, and programs to help meet the client's needs.

Setting up a triage definition

Set up a triage definition as follows:

- Open the XML document that represents the intake process definition - see Appendix A.
- Create or locate an XML element `IndividualConfiguration` as a direct child of the `IntakeProcess` element.

Note: Only one `IndividualConfiguration` element can exist in the document.

- Create or locate an XML element `Triage` as a direct child of the `IndividualConfiguration` element previously located.
- Specify the following attributes on this element to define the Triage process:

Table 11: Triage Attributes

Attribute Name	Type	Purpose
<code>DataStore</code>	String	Specifies the name of the datastore schema to be used for the IEG script that is run to capture the details of the triage.
<code>RuleSet</code>	String	Specifies the name of the CER rule set to be used to process the answers from the script into recommendations for programs and services.
<code>RuleSetClass</code>	String	Specifies the name of the CER rule class in the CER rule set that is used to calculate the recommendations of the triage.
<code>ScriptName</code>	String	Specifies the name of the IEG script that is run to capture the details of the triage.
<code>ScriptType</code>	String	Specifies the type of IEG script that is run to capture the details of the triage.
<code>ScriptVersion</code>	String	Specifies the type of IEG script that is run to capture the details of the triage.

Adding urgent alerts to the rule set

To output an urgent alert from the CER rule set, the rule class that is defined in the triage definition should have an attribute that is named `alertMessages`. This attribute has as its type a list of rule classes of type `AlertMessage`. Each `AlertMessage` that is calculated and passed into this list appears as an urgent alert on the **Triage Result** page.

Recommending services

To recommend a service from the CER rule set, the rule class that is specified in the triage definition should have an attribute that is named `services`. This attribute has as its type a list of rule classes that subclass the `AbstractService` type. Each `AbstractService` that is calculated and passed into this list appears as a recommended service on the **Triage Result** page.

Recommending programs

To recommend a program from the CER rule set, the rule class that is defined as part of triage definition should have an attribute that is named `programs`. This attribute has as its type a list of rule classes that subclass the type `AbstractProgram`. Each `AbstractProgram` that is calculated and passed into this list appears as a recommended program on the **Triage Result** page.

1.4 Defining screenings

Screening allows a client to determine whether they are potentially eligible for one or more programs based on a set of high level, guided questions. Potential eligibility for the selected programs is determined by running eligibility rules against the client's responses to the questions.

Setting up a Screening Definition

- Open the XML document which represents the intake process definition - see Appendix A.
- Create or locate an XML element `IndividualConfiguration` as a direct child of the `IntakeProcess` element. Only one `IndividualConfiguration` element may exist in the document.
- Create or locate an XML element `Screening` as a direct child of the `IndividualConfiguration` element previously located. There can be as many `Screening` elements as are required.
- On this element the following attributes can be specified which are used to define the screening process.

Table 12: Screening Attributes

This table describes the attributes that can be set for a screening definition.

Attribute Name	Type	Purpose
Type	Code Table Entry	This attribute specifies the category of screening. The category is an entry from the <code>ScreeningCategory</code> code table. Only one screening script can be specified for each screening category, and as such this is a unique attribute.
DataStore	String	This attribute specifies the name of the data store schema to be used for the IEG script that is executed to capture the details of the screening.

Attribute Name	Type	Purpose
RuleSet	String	This attribute specifies the name of the CER rule set to be used to process the answers from the screening script into recommendations for programs.
RuleSetClass	String	This attribute specifies the name of the CER rule class in the CER rule set which is used to calculate the recommendations of the screening.
ScriptName	String	This attribute specifies the name of the IEG script which is used to capture the details of the triage.
ScriptVersion	String	This attribute specifies the type of IEG script which will be used to capture the details of the triage.

Specifying whether Programs are Available Internally or Externally

- Open the XML document which represents the intake process definition - see Appendix A.
- Locate the XML element Screening which is to be defined.
- For each program that the intake process will support, create an element ProgramType as a direct child of the Screening element.
- For each program that the intake process will support, create an element ProgramType as a direct child of the Screening element.

Table 13: ProgramType Attributes for a Screening

This table describes the attributes that can be set for a screening definition.

Attribute Name	Type	Purpose
ProgramType	Code Table Entry	This attribute should contain an entry from the code table 'ProgramType'. This is a code table which specifies a program type, e.g. medical assistance. This attribute is mandatory and must be specified.
Availability	String	This attribute should contain the value 'internal' if the program is available internally or 'external' if the program is available from a different agency.

- **DisplayStatus: SCREENINGSTATUS:** The set of statuses for which a screening should be displayed on the list pages. If a status is not listed here then screenings of that status will not be displayed in the user interface.

```
<?xml version="1.0" encoding="UTF-8"?>
  <codetables package="curam.intake.codetable">
    <codetable java_identifier="SCREENINGSTATUS"
      name="SCREENINGSTATUS">
      <displaynames>
```

```

<locale language="en">Screening Status</locale>
</displaynames>
<code default="true" java_identifier="IN_PROGRESS"
  status="ENABLED" value="SS19001">
  <locale language="en" country="US" sort_order="0">
  <description>In-Progress</description>
  <annotation></annotation>
  </locale>
</code>
<code default="false" java_identifier="COMPLETE"
  status="ENABLED" value="SS19002">
  <locale language="en" country="US" sort_order="0">
  <description>Complete</description>
  <annotation></annotation>
  </locale>
</code>
<code default="false" java_identifier="CANCELLED"
  status="ENABLED" value="SS19003">
  <locale language="en" country="US" sort_order="0">
  <description>Cancelled</description>
  <comments>Comments for Rejected in EN_US</comments>
  <annotation></annotation>
  </locale>
</code>
</codetable>
</codetables>

```

Figure 2: For example:

Following are the codes in CT_ScreeningStatus.ctx:

- Now, Please refer to the following example for guidance:

```

<Screening DataStore="InternalApplication"
RuleSet="FinancialAssistanceRuleSet"
RuleSetClass="FinancialAssistance"
  ScriptName="FinancialAssistanceApplicationScreening"
  ScriptVersion="V1" type="SC19001" >
  <ProgramType availability="internal" type="PT19001"/>
  <ProgramType availability="internal" type="PT19002"/>
  <ProgramType availability="internal" type="PT19003"/>
  <DisplayStatus status="SS19002"/>
  <DisplayStatus status="SS19001"/>
</Screening>

```

1.) Create the file CommonIntakeProcess.xml in the <EJBSERVER_HOME>\components\CommonIntake\data\initial\clob" folder.

2.)Following entries need to be made in the DMX files:

Note: If the dmx does not exist, then need to create a dmx file.

```

<row>
  <attribute name="intakeProcessID">
    <value>27900</value>
  </attribute>
  <attribute name="name">
    <value>CommonIntakeProcess</value>
  </attribute>
  <attribute name="state">
    <value>IPDS19002</value>
  </attribute>
</row>

```

```

</attribute>
<attribute name="version">
<value>1</value>
</attribute>
</row>

```

2.1) INTAKEPROCESSNAME.dmx - Entry of the Screening Process Name.

2.2) INTAKEPROCESSCONFIGURATION.dmx - Entry of the process configuration.

```

<row>
  <attribute name="intakeProcessConfigurationID">
    <value>27900</value>
  </attribute>
  <attribute name="intakeProcessNameID">
    <value>27900</value>
  </attribute>
  <attribute name="optionXML">
    <value>./CommonIntake/data/initial/clob/
CommonIntakeProcess.xml</value>
  </attribute>
</row>

```

2.3) INTAKEPROCESSROLEASSIGNMENT.dmx - Entry of the process role assignment.

```

<row>
  <attribute name="intakeProcessRoleAssignmentID">
    <value>27900</value>
  </attribute>
  <attribute name="intakeProcessID">
    <value>27900</value>
  </attribute>
  <attribute name="role">
    <value>CIAPP</value>
  </attribute>
  <attribute name="recordStatus">
    <value>RST1</value>
  </attribute>
</row>

```

2.4) IEGSCRIPTINFO.dmx - Information about where the IEG script is residing.

```

<row>
  <attribute name="scriptID">
    <value>FinancialAssistanceApplicationScreening</value>
  </attribute>
  <attribute name="type">
    <value>Screening</value>
  </attribute>
  <attribute name="scriptVersion">
    <value>V1</value>
  </attribute>
  <attribute name="name">
    <value>FinancialAssistanceApplicationScreening</value>
  </attribute>
  <attribute name="scriptDefinition">

```

```

    <value>./CommonIntake/data/initial/clob/
FinancialAssistanceApplicationScreening_V1_Screening.xml</
value>
  </attribute>
  <attribute name="status">
    <value>ISS02</value>
  </attribute>
  <attribute name="versionNo">
    <value>1</value>
  </attribute>
  <attribute name="lastWritten">
    <value>SYSTIME</value>
  </attribute>
</row>

```

Note: If the dmx does not exist, then need to create a dmx file.

```

<row>
  <attribute name="creoleRuleSetID">
    <value>27011</value>
  </attribute>
  <attribute name="name">
    <value>FinancialAssistanceRuleSet</value>
  </attribute>
  <attribute name="ruleSetDefinition">
    <value>../../build/svr/creole.gen/Rules/components/
CommonIntake/FinancialAssistanceRuleSet.xml</value>
  </attribute>
  <attribute name="ruleSetVersion">
    <value/>
  </attribute>
  <attribute name="versionNo">
    <value>1</value>
  </attribute>
</row>

```

2.5)CREOLERULESET.dmx - Rule set information which is used to determine which products is the user eligible.

```

<row>
  <attribute name="schemaName">
    <value>InternalApplication</value>
  </attribute>
  <attribute name="schemaText">
    <value>./CommonIntake/data/initial/clob/
InternalApplication.xsd</value>
  </attribute>
  <attribute name="versionNo">
    <value>1</value>
  </attribute>
  <attribute name="lastWritten">
    <value>SYSTIME</value>
  </attribute>
</row>
<row>
  <attribute name="schemaName">
    <value>InternalApplicationDomains</value>
  </attribute>
  <attribute name="schemaText">

```

```

    <value>./CommonIntake/data/initial/clob/
InternalApplicationDomains.xsd</value>
  </attribute>
  <attribute name="versionNo">
    <value>1</value>
  </attribute>
  <attribute name="lastWritten">
    <value>SYSTIME</value>
  </attribute>
</row>

```

2.6) DATASTORESCHEMA.dmx - Datastore information which is used by the IEG script.

```

<row>
  <attribute name="schemaName">
    <value>InternalApplication</value>
  </attribute>
  <attribute name="type">
    <value>DSST1</value>
  </attribute>
  <attribute name="localeIdentifier">
    <value language="en">en</value>
  </attribute>
  <attribute name="localizedText">
    <value language="en">The Internal Applications Schema</
value>
  </attribute>
  <attribute name="versionNo">
    <value>1</value>
  </attribute>
  <attribute name="lastWritten">
    <value>SYSTIME</value>
  </attribute>
</row>
<row>
  <attribute name="schemaName">
    <value>InternalApplicationDomains</value>
  </attribute>
  <attribute name="type">
    <value>DSST1</value>
  </attribute>
  <attribute name="localeIdentifier">
    <value language="en">en</value>
  </attribute>
  <attribute name="localizedText">
    <value language="en">The Internal Applications Domains
Schema</value>
  </attribute>
  <attribute name="versionNo">
    <value>1</value>
  </attribute>
  <attribute name="lastWritten">
    <value>SYSTIME</value>
  </attribute>
</row>

```

2.7) DATASTORESCHEMATRANSLATION.dmx - Data store schema translation information

```

<row>
  <attribute name="resourceid">
    <value>19431</value>
  </attribute>
  <attribute name="localeIdentifier">
    <value/>
  </attribute>
  <attribute name="name">
    <value>financialassistance_about_you.png</value>
  </attribute>
  <attribute name="contentType">
    <value>image/png</value>
  </attribute>
  <attribute name="contentDisposition">
    <value>inline</value>
  </attribute>
  <attribute name="content">
    <value>./CommonIntake/data/initial/blob/
financialassistance_about_you.png</value>
  </attribute>
  <attribute name="internal">
    <value>0</value>
  </attribute>
  <attribute name="lastWritten">
    <value>SYSTIME</value>
  </attribute>
  <attribute name="versionNo">
    <value>1</value>
  </attribute>
  <attribute name="category">
    <value>RS_IMG</value>
  </attribute>
</row>
<row>
  <attribute name="resourceid">
    <value>19432</value>
  </attribute>
  <attribute name="localeIdentifier">
    <value/>
  </attribute>
  <attribute name="name">
    <value>financialassistance_title_about_you.png</value>
  </attribute>
  <attribute name="contentType">
    <value>image/png</value>
  </attribute>
  <attribute name="contentDisposition">
    <value>inline</value>
  </attribute>
  <attribute name="content">
    <value>./CommonIntake/data/initial/blob/
financialassistance_title_about_you.png</value>
  </attribute>
  <attribute name="internal">
    <value>0</value>
  </attribute>
  <attribute name="lastWritten">

```

```

    <value>SYSTIME</value>
  </attribute>
  <attribute name="versionNo">
    <value>1</value>
  </attribute>
  <attribute name="category">
    <value>RS_IMG</value>
  </attribute>
</row>
<row>
  <attribute name="resourceid">
    <value>19433</value>
  </attribute>
  <attribute name="localeIdentifier">
    <value/>
  </attribute>
  <attribute name="name">
    <value>Application Form.pdf</value>
  </attribute>
  <attribute name="contentType">
    <value>image/png</value>
  </attribute>
  <attribute name="contentDisposition">
    <value>inline</value>
  </attribute>
  <attribute name="content">
    <value>./CommonIntake/data/initial/blob/Application
Form.pdf</value>
  </attribute>
  <attribute name="internal">
    <value>0</value>
  </attribute>
  <attribute name="lastWritten">
    <value>SYSTIME</value>
  </attribute>
  <attribute name="versionNo">
    <value>1</value>
  </attribute>
  <attribute name="category">
    <value/>
  </attribute>
</row>
<row>
  <attribute name="resourceID">
    <value>19434</value>
  </attribute>
  <attribute name="localeIdentifier">
    <value/>
  </attribute>
  <attribute name="name">
    <value>FinancialAssistanceApplication_V1_Intake</value>
  </attribute>
  <attribute name="contentType">
    <value>text/plain</value>
  </attribute>
  <attribute name="contentDisposition">
    <value>inline</value>
  </attribute>

```

```

    <attribute name="content">
    <value>./CommonIntake/data/initial/blob/
financialAssApplication.properties</value>
    </attribute>
    <attribute name="internal">
    <value>0</value>
    </attribute>
    <attribute name="lastWritten">
    <value>SYSTIME</value>
    </attribute>
    <attribute name="versionNo">
    <value>1</value>
    </attribute>
    <attribute name="category">
    <value>RS_PROP</value>
    </attribute>
</row>
<row>
    <attribute name="resourceid">
    <value>19435</value>
    </attribute>
    <attribute name="localeIdentifier">
    <value/>
    </attribute>
    <attribute name="name">
    <value>financial-ass-app-config</value>
    </attribute>
    <attribute name="contentType">
    <value>text/plain</value>
    </attribute>
    <attribute name="contentDisposition">
    <value>inline</value>
    </attribute>
    <attribute name="content">
    <value>./CommonIntake/data/initial/blob/financial-ass-app-
config.properties</value>
    </attribute>
    <attribute name="internal">
    <value>0</value>
    </attribute>
    <attribute name="lastWritten">
    <value>SYSTIME</value>
    </attribute>
    <attribute name="versionNo">
    <value>1</value>
    </attribute>
    <attribute name="category">
    <value>RS_PROP</value>
    </attribute>
</row>
<row>
    <attribute name="resourceID">
    <value>19436</value>
    </attribute>
    <attribute name="localeIdentifier">
    <value/>
    </attribute>
    <attribute name="name">

```

```

<value>FinancialAssistanceApplication_V1_Intake_AboutTheClaimantContinu
value>
  </attribute>
  <attribute name="contentType">
    <value>text/plain</value>
  </attribute>
  <attribute name="contentDisposition">
    <value>inline</value>
  </attribute>
  <attribute name="content">
    <value>./CommonIntake/data/initial/blob/
AboutTheClaimantContinue.properties</value>
  </attribute>
  <attribute name="internal">
    <value>0</value>
  </attribute>
  <attribute name="lastWritten">
    <value>SYSTIME</value>
  </attribute>
  <attribute name="versionNo">
    <value>1</value>
  </attribute>
  <attribute name="category">
    <value>RS_PROP</value>
  </attribute>
</row>
<row>
  <attribute name="resourceID">
    <value>19437</value>
  </attribute>
  <attribute name="localeIdentifier">
    <value/>
  </attribute>
  <attribute name="name">

  <value>FinancialAssistanceApplicationScreening_V1_Screening_AboutTheCla
value>
  </attribute>
  <attribute name="contentType">
    <value>text/plain</value>
  </attribute>
  <attribute name="contentDisposition">
    <value>inline</value>
  </attribute>
  <attribute name="content">
    <value>./CommonIntake/data/initial/blob/
AboutTheClaimantContinue.properties</value>
  </attribute>
  <attribute name="internal">
    <value>0</value>
  </attribute>
  <attribute name="lastWritten">
    <value>SYSTIME</value>
  </attribute>
  <attribute name="versionNo">
    <value>1</value>
  </attribute>

```

```

    <attribute name="category">
    <value>RS_PROP</value>
    </attribute>
  </row>
  <row>
    <attribute name="resourceID">
    <value>19439</value>
    </attribute>
    <attribute name="localeIdentifier">
    <value/>
    </attribute>
    <attribute name="name">

    <value>FinancialAssistanceApplicationScreening_V1_Screening</
value>
    </attribute>
    <attribute name="contentType">
    <value>text/plain</value>
    </attribute>
    <attribute name="contentDisposition">
    <value>inline</value>
    </attribute>
    <attribute name="content">
    <value>./CommonIntake/data/initial/blob/
financialAssApplicationScreening.properties</value>
    </attribute>
    <attribute name="internal">
    <value>0</value>
    </attribute>
    <attribute name="lastWritten">
    <value>SYSTIME</value>
    </attribute>
    <attribute name="versionNo">
    <value>1</value>
    </attribute>
    <attribute name="category">
    <value>RS_PROP</value>
    </attribute>
  </row>
  <row>
    <attribute name="resourceID">
    <value>19440</value>
    </attribute>
    <attribute name="localeIdentifier">
    <value/>
    </attribute>
    <attribute name="name">

    <value>curam.humanservicesscreening.rules.FinancialAssistanceInternalSc
value>
    </attribute>
    <attribute name="contentType">
    <value>text/plain</value>
    </attribute>
    <attribute name="contentDisposition">
    <value>inline</value>
    </attribute>
    <attribute name="content">

```

```

    <value>./CommonIntake/data/initial/blob/
FinancialAssistanceInternalScreening.properties</value>
    </attribute>
    <attribute name="internal">
    <value>0</value>
    </attribute>
    <attribute name="lastWritten">
    <value>SYSTIME</value>
    </attribute>
    <attribute name="versionNo">
    <value>1</value>
    </attribute>
    <attribute name="category">
    <value>RS_PROP</value>
    </attribute>
</row>
<row>
    <attribute name="resourceID">
    <value>19441</value>
    </attribute>
    <attribute name="localeIdentifier">
    <value/>
    </attribute>
    <attribute name="name">
    <value>CIIntakeProcess.GlobalConfiguration</value>
    </attribute>
    <attribute name="contentType">
    <value>text/plain</value>
    </attribute>
    <attribute name="contentDisposition">
    <value>inline</value>
    </attribute>
    <attribute name="content">
    <value>./CommonIntake/data/initial/blob/
CIIntakeProcessGlobalConfiguration.properties</value>
    </attribute>
    <attribute name="internal">
    <value>0</value>
    </attribute>
    <attribute name="lastWritten">
    <value>SYSTIME</value>
    </attribute>
    <attribute name="versionNo">
    <value>1</value>
    </attribute>
    <attribute name="category">
    <value>RS_PROP</value>
    </attribute>
</row>
<row>
    <attribute name="resourceID">
    <value>1942</value>
    </attribute>
    <attribute name="localeIdentifier">
    <value/>
    </attribute>
    <attribute name="name">

```

```

<value>CIIntakeProcess.ApplicationConfiguration.AT19001.RightsAndRespon
value>
  </attribute>
  <attribute name="contentType">
    <value>text/plain</value>
  </attribute>
  <attribute name="contentDisposition">
    <value>inline</value>
  </attribute>
  <attribute name="content">
    <value>./CommonIntake/data/initial/blob/
RightsAndResponsibilitiesText.properties</value>
  </attribute>
  <attribute name="internal">
    <value>0</value>
  </attribute>
  <attribute name="lastWritten">
    <value>SYSTIME</value>
  </attribute>
  <attribute name="versionNo">
    <value>1</value>
  </attribute>
  <attribute name="category">
    <value>RS_PROP</value>
  </attribute>
</row>
<row>
  <attribute name="resourceID">
    <value>19443</value>
  </attribute>
  <attribute name="localeIdentifier">
    <value/>
  </attribute>
  <attribute name="name">

  <value>CIIntakeProcess.ApplicationConfiguration.AT19001.AuthorizationIn
value>
  </attribute>
  <attribute name="contentType">
    <value>text/plain</value>
  </attribute>
  <attribute name="contentDisposition">
    <value>inline</value>
  </attribute>
  <attribute name="content">
    <value>./CommonIntake/data/initial/blob/
AuthorizationInformationText.properties</value>
  </attribute>
  <attribute name="internal">
    <value>0</value>
  </attribute>
  <attribute name="lastWritten">
    <value>SYSTIME</value>
  </attribute>
  <attribute name="versionNo">
    <value>1</value>
  </attribute>

```

```

    <attribute name="category">
    <value>RS_PROP</value>
    </attribute>
</row>
<row>
    <attribute name="resourceID">
    <value>19444</value>
    </attribute>
    <attribute name="localeIdentifier">
    <value/>
    </attribute>
    <attribute name="name">
    <value>CIIntakeProcess.ApplicationConfiguration.AT19001.DepartmentPolicy
value>
    </attribute>
    <attribute name="contentType">
    <value>text/plain</value>
    </attribute>
    <attribute name="contentDisposition">
    <value>inline</value>
    </attribute>
    <attribute name="content">
    <value>./CommonIntake/data/initial/blob/
DepartmentPolicyText.properties</value>
    </attribute>
    <attribute name="internal">
    <value>0</value>
    </attribute>
    <attribute name="lastWritten">
    <value>SYSTIME</value>
    </attribute>
    <attribute name="versionNo">
    <value>1</value>
    </attribute>
    <attribute name="category">
    <value>RS_PROP</value>
    </attribute>
</row>

```

2.8) APPRESOURCE.dmx - Entries of various Appresource properties.

Make sure that the files mentioned in the "content" attribute is present in the location specified.

Here it is "CommonIntake/data/initial/blob/" folder

3) Following files need to be copied into clob folder:

Note: If the files do not exist then need to create a new files.

a) Screening IEG questionnaire script

For ex: Copy FinancialAssistanceApplicationScreening_V1_Screening.xml to "CommonIntake\data\initial\clob" folder

b) XSD information

For ex: Copy InternalApplication.xsd to "CommonIntake\data\initial\clob" folder

Copy InternalApplicationDomains.xsd to "CommonIntake\data\initial\clob" folder

c) Process Name XML -

For ex: Copy existing CommonIntakeProcess.xml to "CommonIntake\data\initial\clob" folder

d) Rule set XML -

For ex: Copy FinancialAssistanceRuleSet.xml to "CommonIntake\CREOLE_Rule_Sets" folder

All the files here have been copied from Intake stream to make Screening configuration.

Adding urgent alerts to the rule set

To output an urgent alert from the CER rule set, the rule class defined in the screening definition should have an attribute named .

The `alertMessages` attribute has as its type a list of rule classes of type `AlertMessage`. Each `AlertMessage` which is calculated for this list appears as an urgent alert on the **Triage Result** page in the application.

Recommending programs

To recommend a program from the CER rule set, the rule class that is defined in the screening definition has an attribute named `programs`. This attribute has as its type a list of rule classes that subclass the type `AbstractProgram`. Each `AbstractProgram` is calculated into this list appears as a recommended program on the **screening result** page.

Specifying whether programs are available internally or externally

Each program that is output from the screening can be available internally or externally. If the program is available internally, then a link to start the application script is displayed. If the program is available externally, then a link to refer the client to the external agency is displayed. Screening results behavior can be defined to specify whether specific programs are available internally or externally as follows:

- Open the XML document that represents the intake process definition.
- Locate the XML element *Screening* is to be defined.
- For each program that the intake process supports, create an element *ProgramType* as a direct child of the *Screening* element.

Table 14: ProgramType Attributes for a Screening

Attribute Name	Type	Purpose
<i>ProgramType</i>	Code Table Entry	Contains an entry from the code table ' <i>ProgramType</i> '. This is a code table that specifies a program type, for example, medical assistance. This attribute is mandatory.
<i>Availability</i>	String	Contains the value ' <i>internal</i> ' if the program is available internally or ' <i>external</i> ' if the program is available from a different agency.

1.5 Configuring client applications

To file a claim for benefits for most needs-based programs, a client must submit an application, providing required information on personal details, income, expenses, employment, education etc.

When an application is submitted, there are a number of processing steps that are carried out either manually by the worker, or automatically by the system, depending on the agency and the program. Each of these steps is configurable. Application routing configuration options are also available.

Setting up the application configuration

- Open the XML document that represents the Intake Process Configuration.
- Create or locate an XML element *ApplicationConfiguration* as a direct child of the *IntakeProcess*, or as a root element. There can only be one *ApplicationConfiguration* element per document.
- For each application to be handled by the Intake Process, create an XML element *ApplicationType* as a direct child of the *ApplicationConfiguration* element previously located. There can be as many *ApplicationType* elements specified as required.
- On this element the following attributes can be specified which configures the Application process for that application type:

Table 15: *ApplicationType* Attributes for an Application

Attribute Name	Type	Purpose
Type	Code Table Entry	This attribute specifies the type of application that this element represents. The type is an entry from the <i>ApplicationType</i> code table. Only one application script can be specified for each <i>ApplicationType</i> , and is a unique attribute.
ShowPreviewPanel	String	This attribute specifies whether a preview panel is displayed for this type of application on the Application List Page in Common Intake.

Configuring script details

- Open the XML document that represents the intake process configuration.
- Locate the XML element *ApplicationType*, which represents the application type to be configured.
- Create or locate an XML element *ApplicationScript* as a direct child of the *ApplicationType* element.
- On this element, the following attributes can be specified which configures the application script for that application type:

Table 16: ApplicationScript Attributes for an Application

Attribute Name	Type	Purpose
<i>SchemaName</i>	String	Specifies the name of the data store schema to be used for the IEG script that is run to capture the details of the application.
<i>ScriptName</i>	String	Specifies the name of the IEG Script that is used to capture the details of the application.
<i>ScriptVersion</i>	String	Specifies which version of the named IEG script to use to capture the details of the application.
<i>SimpleScriptName</i>	String	Specifies the name of the IEG script that is used to capture the details of the application. This differs from the <i>ScriptName</i> attribute, in that it is used where a case exists, while the script specified in the <i>ScriptName</i> attribute is used when no existing case exists. This might not be the same script that is specified in the <i>ScriptName</i> attribute and is present to allow more flexibility of configuration.

Configuring the programs covered by an application type

Configure the program type attributes, configure program mapping, and add a timer for the program.

Configure program type attributes

- Open the XML document which represents the intake process configuration.
- Locate the XML element *ApplicationType* which represents the application type to be configured.
- Create or locate an XML element *ProgramType* as a direct child of the *ApplicationType* element.
- On this element the following attributes can be specified which configure the application script for that application type.

Table 17: ProgramType Attributes

Attribute Name	Type	Purpose
<i>type</i>	Code Table Entry	An entry from the program type code table.

Attribute Name	Type	Purpose
<i>isReopenAllowed</i>	Boolean	Controls whether the reopen program option will be available in the row action menu on the Programs list page. If set to true the "Reopen" action is enabled for denied or withdrawn program applications. If set to false the "Reopen" action is disabled.
<i>isResumeTimerEnabled</i>	Boolean	Controls whether the whether a program application timer resumes on reopening a denied or withdrawn program application. If set to true the timer resumes on reopening. If set to false the timer does not resume on reopening.
MaxNumberReopeningsAllowed	Integer	Specifies the number of times it is possible to reopen an application.
TimerRestartDate	Code Table Entry	Specifies the type of date that is used to resume the timer. The date type is an entry from the <i>TimerRestartDate</i> code table.

Configuring the program mapping

- Create or locate an XML element *ProgramMapping* as a direct child of the *ProgramType* element.
- On the *ProgramMapping* element the following attributes can be specified.

Table 18: *ProgramMapping* Attributes

Attribute Name	Type	Purpose
<i>IntegratedCaseType</i>	Code Table Entry	Specifies the type of Integrated Case to create if this Program Type was applied for.
<i>MappingXML</i>	String	Specifies the Mapping XML file to be used to map the data from the script into evidence on the integrated case that is created.
<i>MappingXMLConfiguration</i>	String	Specifies the mapping configuration XML file to be used to map the data from the script into evidence on the integrated case that is created.

Adding a timer for the program

The timer functionality in Common Intake is based upon milestones functionality. A 'timer' milestone can be configured to be started when an application for a program is submitted. To achieve this:

- Create or locate an XML element Milestone as a direct child of the ProgramType element.

- On the Milestone element the following attributes can be specified:

Table 19: Milestone Attributes

Attribute Name	Type	Purpose
<i>ID</i>	Long	Specifies the primary key of the milestone to be created when this particular program is applied for.

Note: A milestone configuration record contains an Earliest Start Day field (specified in days) to support functionality for case and service plan milestones. This field is defined as "The shortest time in days that this milestone can start after the service plan or case start date.". If is it set to anything other than 0 for an application program timer it has the effect of extending the timer by the specified number of days.

Integrating the application with Universal Access

Applications can be received either by the caseworker in Common Intake, or by using Universal Access. Common Intake can be configured so that when an application is received from Universal Access the application becomes a Common Intake application and can be processed by the case worker in the same way as if it had been captured by using Common Intake.

To configure Common Intake to pick up an application from Universal Access take the following steps:

- Open the XML document that represents the intake process configuration.
- Locate the XML element *ApplicationType*, which represents the application type to be configured.
- Create or locate an XML element *IntakeApplicationType* as a direct child of the *ApplicationType* element. Only one *IntakeApplicationType* element might be specified per *ApplicationType*.
- Specify the attribute that is described in table 1 in the *IntakeApplicationType* element:

Table 20: IntakeApplicationType Attributes

This table describes the attributes for the IntakeApplicationType element.

Attribute Name	Type	Purpose
<i>MappedType</i>	Long	This attribute specifies the primary key of a record in the <i>IntakeApplicationType</i> database table in Workspace Services. With this configured, Common Intake automatically picks up applications that are based on this <i>IntakeApplicationType</i> and present it as an application for processing to the caseworker .

Configuring workflows for application processing

Common Intake supports the configuration of workflows for the assignment of applications and related tasks to workers. Workflows can be configured to reflect the workflow process the organization requires.

Common Intake ships with three basic starter workflows, which can be used to get up and running quickly.

To configure a workflow to be used for a particular application, take the following steps:

- Open the XML document, which represents the intake process configuration.
- Locate the XML element *ApplicationType* represents the application type to be configured.
- Create or locate an XML element *Workflows* as a direct child of the *ApplicationType* element. Only one workflow element can be specified per *ApplicationType*.
- Create or locate an XML element *OnlineApplicationRecieved* as a direct child of the *Workflows* element. This has one supported attribute name, which is the name of the workflow to be enacted when an online application is received from UA.
- Create or locate an XML element *WithdrawalRequest* as a direct child of the *Workflows* element. This has one supported attribute name, which is the name of the workflow to be enacted when a withdrawal request is received from UA.
- Create or locate an XML element *ReadyForDetermination* as a direct child of the *Workflows* element. This has one supported attribute name, which is the name of the workflow to be enacted when the application transitions to a status of ready for determination.
- Create or locate an XML element *ReassignApplicationNotification* as a direct child of the *Workflows* element. This has one supported attribute, name, which is the name of the workflow to be enacted when the application owner is reassigned.

Configuring an application workflow allocation target

Each of the workflow elements, *OnlineApplicationRecieved*, *WithdrawalRequest*, and *ReadyForDetermination* has a child element of type *AllocationTarget*. To configure the allocation target of one of the application workflows, take the following steps:

- Open the XML document, which represents the intake process configuration.
- Locate the XML element *ApplicationType* represents the application type to be configured.
- Locate an XML element *Workflow* as a direct child of the *ApplicationType* element.
- Locate one of the workflow elements detailed above.
- Create or locate an XML element *AllocationTarget*.
- On the element, *AllocationTarget* the following attributes can be specified:

Table 21: AllocationTarget Attributes

Attribute Name	Type	Purpose
<i>Type</i>	Code Table Entry	An entry from the target item type code table. Supported target item types are: work queue and user.

Attribute Name	Type	Purpose
<i>Name</i>	String	If the type attribute is set to user, this attribute should specify the user's name. Alternatively if the type is set to work queue this attribute should contain the primary key of the database table WorkQueue.

Configuring the texts associated with an application

Change the text that is included in an application and communicated to an individual before the application can be submitted.

Rights and responsibilities text

Each application that is run by using **Common Intake** must have a rights and responsibilities text specified. This information is the text that appears in the application before the caseworker submits the application script. The caseworker must confirm that the text is communicated to the individual applying for the program before continuing the application process.

To configure the rights and responsibilities text, complete the following tasks:

- Create a property file resource in the resource store with the following name:
IntakeProcessName.ApplicationConfiguration.*ApplicationType*.RightsAndResponsibilitiesText
- Replace *IntakeProcessName* with the logical name of the intake process as stored in the *IntakeProcessName* database table.
- Replace *ApplicationType* with the code table code of the application type that this rights and responsibilities text is for.
- Create a property in the property file with the name DisplayText and specify the text to be displayed as the value of the property.

Note: The value of the property can contain HTML to format the display text and provide a rich text feel.

Authorization information text

Authorization information text appears in the application before the caseworker submits the application script. The caseworker is required to confirm that the text is communicated to the individual applying for the program before the caseworker continues the application process.

This text is configured in an identical fashion to the rights and responsibilities text except that the name of the property file resource is different. In this case, the property file needs to be named:

- *IntakeProcessName*.ApplicationConfiguration.*ApplicationType*.AuthorizationInformationText
- Replace *IntakeProcessName* with the logical name of the intake process as stored in the *IntakeProcessName* database table.
- Replace *ApplicationType* with the code table code of the application type that this rights and responsibilities text is for.

Department policy text

Department policy text appears in the application before the caseworker submits the application script. The caseworker must confirm that the text is communicated to the individual applying for the program before the caseworker continues with submission of the application.

This text is configured in an identical fashion to the rights and responsibilities text except that the name of the property file resource is different. In this case, the property file needs to be named:

- `IntakeProcessName.ApplicationConfiguration.ApplicationType.DepartmentPolicyText`
 - Replace `IntakeProcessName` with the logical name of the Intake Process as stored in the `IntakeProcessName` database table.
 - Replace `ApplicationType` with the code table code of the Application Type that this rights and responsibilities text is for.

Configuring cases to which the application can be transferred

Configure the cases that the application can be transferred to.

Use the following steps to configure the cases:

- Open the XML document that represents the intake process configuration.
- Locate the XML element `ApplicationType` that represents the application type to be configured.
- Create or locate an XML element `TransferToExistingCase` as a direct child of the `ApplicationType` element.

Configuring Case Types

The types of cases that the application can be transferred to can be configured as follows:

- For each case type that you want to allow the application to be transferred to, create an element `CaseType` as a direct child of `TransferToExistingCase`.
- On this element, the following attribute can be specified which configures the type of case that an application can be transferred to:

Table 22: CaseType Attributes

Attribute Name	Type	Purpose
<code>Type</code>	Code Table Entry	Contains an entry from the case type code table.

Configuring case statuses

The statuses of cases that the application can be transferred to can be configured as follows:

- For each case status that you want to allow the application to be transferred to, create an element `CaseStatus` as a direct child of `TransferToExistingCase`.
- On this element, the following attribute can be specified which configures the status of case that an application can be transferred to:

Table 23: CaseStatus Attributes

Attribute Name	Type	Purpose
Status	Code Table Entry	Contains an entry from the case status code table.

Configuring application statuses

The statuses of the applications that can be transferred can be configured as follows:

- For each application status that you want to transfer, create an element *ApplicationStatus* as a direct child of *TransferToExistingCase*.
- On this element, the following attribute can be specified which configures the status of application that can be transferred:

Table 24: ApplicationStatus Attributes

Attribute Name	Type	Purpose
Status	Code Table Entry	Contains an entry from the application status code table.

Configuring the cases view for the application

The application workspace contains a **Cases** tab that displays cases that are associated with the application.

Certain information on the **Cases** tab is configurable. To configure the cases view for the application, do the following:

- Open the XML document, which represents the intake process configuration.
- Locate the XML element *ApplicationType* represents the application type to be configured.
- Create or locate an XML element *ShowExistingCase* as a direct child of the *ApplicationType* element.

Configuring case preview panels

Preview panels for the cases in the list can be configured based on case type as follows:

- Create or locate an XML element *PreviewPanel* as a direct child of the *ShowExistingCase* element.
- On this element, the following attribute can be specified which configures the page to display in the preview panel for records on the application cases list:

Table 25: PreviewPanel Attributes

Attribute Name	Type	Purpose
Page	String	Contains the name of the client page that is used for the preview panel.

Configuring case types

The types of cases that are displayed on the list can be configured as follows:

- For each case type you want to display on the application cases list, create an element `ShowCasesOfType` as a direct child of `ShowExistingCase`.
- On this element, the following attribute can be specified which configures the type of cases to be displayed on the application cases list:

Table 26: *ShowCasesOfType* Attributes

Attribute Name	Type	Purpose
<i>Type</i>	Code Table Entry	Contains an entry from the case type code table.

Configuring case statuses

The statuses of cases that are displayed on the list can be configured as follows:

- For each case status you want to display on the application cases list, create an element `ShowCasesOfStatus` as a direct child of `ShowExistingCase`.
- On this element, the following attribute can be specified which configures the status of cases to be displayed on the application cases list:

Table 27: *ShowCasesOfStatus* Attributes

Attribute Name	Type	Purpose
<i>Status</i>	Code Table Entry	Contains an entry from the case status code table.

Configuring undisposed applications

Whether cases with undisposed applications should be displayed can be configured as follows:

- Create an element `ShowCasesWithUndisposedApplication` as a direct child of `ShowExistingCase`.
- On this element, the following attribute can be specified which configures whether cases with undisposed applications is displayed on the application cases list.

Table 28: *ShowCasesWithUndisposedApplications* Attributes

Attribute Name	Type	Purpose
<i>Value</i>	Boolean	Controls whether cases with undisposed applications are displayed on the individual cases list.

Configuring the cases to which the application can be added

Configure the cases that an application can be added to.

To configure the cases that the application can be added to, do the following:

- Open the XML document, which represents the intake process configuration.
- Locate the XML element `ApplicationType`, which represents the application type to be configured.
- Create or locate an XML element `AddToExistingCase` as a direct child of the `ApplicationType` element.

Configuring case types

The types of cases that the application can be added to can be configured as follows:

- For each case type that you want to allow the application to be added to, create an element *AddToCaseOfType* as a direct child of *AddToExistingCase*.
- On this element, the following attribute can be specified that configures the type of case that an application can be added to:

Table 29: *AddToCaseOfType* Attributes

This table describes configuration attributes for the types of cases that the application can be added to.

Attribute Name	Type	Purpose
<i>Type</i>	Code Table Entry	Contains an entry from the case type code table.

Configuring case statuses

The statuses of cases that the application can be added to can be configured as follows:

- For each case status, you want to allow the application to be added to create an element *AddToCaseOfStatus* as a direct child of *AddToExistingCase*.
- On this element, the following attribute can be specified which configures the status of case that an application can be added to:

Table 30: *AddToCaseOfStatus* Attributes

Attribute Name	Type	Purpose
<i>Status</i>	Code Table Entry	Contains an entry from the case status code table.

Configuring the case status after the application has been transferred

Configure what the status of the case is after the application has been transferred.

To configure the case status after the application had been transferred do the following:

- Open the XML document, which represents the intake process configuration.
- Locate the XML element *ApplicationType*, which represents the application type to be configured.
- Create or locate an XML element *CaseStatusAfterTransfer* as a direct child of the *ApplicationType* element.
- On this element, the following attribute can be specified which configures the status the case should take after the application is transferred:

Table 31: *CaseStatusAfterTransfer* Attributes

Attribute Name	Type	Purpose
<i>status</i>	Code Table Entry	Contains an entry from the case status code table.

Configuring case types

The types of cases that take the new status after the application is transferred can be configured as follows:

- For each case type whose status you want to update after application transfer, create an element *CaseType* as a direct child of *CaseStatusAfterTransfer*.
- On this element, the following attribute can be specified that configures the type of case whose status changes:

Table 32: *CaseType* Attributes

Attribute Name	Type	Purpose
<i>status</i>	Code Table Entry	This attribute contains an entry from the case type code table.

Configuring the case status after undoing application transfer

Configure the status of the case after the application transfer is undone.

To configure the case status, do the following:

- Open the XML document, which represents the intake process configuration.
- Locate the XML element *ApplicationType*, which represents the application type to be configured.
- Create or locate an XML element *CaseStatusAfterUndoTransfer* as a direct child of the *ApplicationType* element.
- On this element, the following attribute can be specified which configure the status the case should take after the application transfer has been undone:

Table 33: *CaseStatusAfterUndoTransfer* Attributes

Attribute Name	Type	Purpose
<i>status</i>	Code Table Entry	Contains an entry from the case status code table.

Configuring case types

The types of cases that take the new status after an application transfer is undone can be configured as follows:

- For each case type whose status you want to update after an application transfer is undone, create an element *CaseType* as a direct child of *CaseStatusAfterUndoTransfer*.
- On this element the following attribute can be specified which configures the type of case whose status changes:

Table 34: *CaseType* Attributes

Attribute Name	Type	Purpose
<i>status</i>	Code Table Entry	Contains an entry from the case type code table.

Configuring the application TODO list context

The application TODO list displays the outstanding tasks that the worker must complete to finish processing the application. You can configure the name of the context that is used to load the TODO list on the application home page.

To configure the list context, do the following:

- Open the XML document, which represents the intake process configuration.
- Locate the XML element *ApplicationType*, which represents the application type to be configured.
- Create or locate an XML element *TODOListAdviceContextKeyName* as a direct child of the *ApplicationType* element.
- On this element, the following attribute can be specified which configures the name of the TODO list context:

Table 35: *TODOListAdviceContextKeyName* Attributes

This table describes configuration attributes for name of the application TODO list context.

Attribute Name	Type	Purpose
name	String	Contains the name of the application TODO list context.

Configuring the application quick search

Configure the results of the quick search for applications for the intake process.

To configure the application quick search results, do the following:

- Open the XML document, which represents the intake process configuration.
- Locate the XML element *ApplicationType* represents the application type to be configured.
- Create or locate an XML element *QuickSearch* as a direct child of the *ApplicationType* element.

Configuring application quick search preview panels

Preview panels for the applications in the results list can be configured as follows:

- For each preview panel, you want to configure on the search create an element *PreviewPanel* as a direct child of *QuickSearch*.
- On this element, the following attributes can be specified which configure the page to display in the quick search results preview panel:

Table 36: *PreviewPanel* Attributes

Attribute Name	Type	Purpose
<i>Page</i>	String	Contains the name of the client page that is used for the preview panel.

Configuring the application statuses that display the preview panel

The statuses of applications that the parent preview panel are used for can be configured as follows:

- For each application status, you want the parent preview panel to be used for create an element *ApplicationStatus* as a direct child of *PreviewPanel*.
- On this element, the following attribute can be specified which configures the status of application that uses the parent preview panel:

Table 37: ApplicationStatus Attributes

Attribute Name	Status	Purpose
<i>status</i>	Code Table Entry	Contains an entry from the application status code table.

1.6 Common intake application properties

The common intake application properties are administered through the system administration application.

Table 38: Common Intake Application Properties

Property Name	Description	Default
<i>curam.case.product.registrars</i>	List of product registrars.	<i>curam.citizen.datahub.holdingcase.holdingevidence.fact.HoldingEvidenceRegistrarFactory, curam.sample.sl.fact.SampleSportingGrantEvidenceRegistrarFactory, curam.core.sl.EvidenceRegistrarFactory</i>
<i>curam.applicationsearch.applicationsearch</i>	Whether intake applications are included in the application banner search.	YES
<i>curam.application.applicationtransfer</i>	Whether the application transfer is enabled.	YES
<i>curam.address.intakeprospectperson.registration.locale</i>	Registration locale for the address data while registering a prospect person when creating an address for which no details are provided.	en_US
<i>curam.intake.map.default.zoom.level</i>	This property is used to configure the initial zoom level of the Google Maps display on the Service Provider screen. Zoom levels between 0 (the lowest zoom level, in which the entire world can be seen on one map) to 21+ (down to individual buildings) are possible.	11
<i>curam.intake.map.default.center.latitude</i>	This property is used to configure the default latitude of the center of the Google Maps display on the Service Provider screen.	-89.40570831298828

Property Name	Description	Default
<i>curam.intake.bom.milestoneconfiguration</i>	When specified to true, this flag ensures that start date of milestone configuration entity is set to current date while transporting.	true
<i>curam.assignment.of.application.to.work.queue</i>	Indicates whether an application can be assigned to a work queue. This variable is used along with the user's Intake Configuration, so if this variable is enabled and the user's allocation target type is a work queue, then the application is routed to the work queue.	YES
<i>curam.application.update.person.details</i>	Indicates whether a person's participant details should be updated with information that is captured in the application script.	NO
<i>curam.application.person.contact.details</i>	Indicates whether the contact details of a person who is captured by using the application script should be set to the primary details of that type, for example, phone number.	NO

Notices

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

Applicability

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

Personal use

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

Commercial use

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

Rights

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

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

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

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

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

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

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

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

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

COPYRIGHT LICENSE:

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

Privacy policy

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

Trademarks

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

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

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

Oracle and Java are registered trademarks of Oracle and/or its affiliates.

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

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

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

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