



# **Cúram 8.1.3**

**Adult Social Care Accelerator Guide**



## Note

---

Before using this information and the product it supports, read the information in [Notices on page 151](#)



# Edition

---

This edition applies to Cúram 8.1, 8.1.1, 8.1.2, and 8.1.3.

© 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

---

<b>Note</b> .....	<b>iii</b>
<b>Edition</b> .....	<b>v</b>
<b>1 Adult Social Care accelerator</b> .....	<b>9</b>
1.1 Overview of Adult Social Care accelerator.....	9
The Adult Social Care business scenario.....	9
Adult Social Care example requirements.....	9
The Adult Social Care business process.....	10
Adult Social Care design decisions.....	11
1.2 Installing the Adult Social Care accelerator.....	11
Prerequisites.....	12
Installing and building the accelerator.....	12
Starting the XML server.....	13
1.3 Performing the Adult Social Care case worker scenario.....	13
Adding a person.....	13
Applying for Adult Social Care.....	14
Authorizing the Adult Social Care application.....	19
Activating the product delivery case.....	19
Generating a payment.....	19
Performing the Adult Social Care workflow.....	20
Capturing changes in circumstance.....	21
1.4 Analyzing the Adult Social Care application.....	25
Completing a fit gap analysis.....	25
Analyzing the organization structure.....	29
Analyzing the user requirements.....	34
Analyzing the intake process requirements.....	36
Analyzing case management.....	41
1.5 Building the Adult Social Care application.....	54
Configuring the component order for Adult Social Care.....	54
Create a user and associated user application view.....	55
Configuring evidence.....	75
Configuring products.....	82
Creating the IEG files for the intake process.....	101
Configuring the intake process.....	104
Creating an Evidence Management wizard.....	118
Modifying person match.....	127
Configuring workflows.....	128

<b>Notices.....</b>	<b>151</b>
Privacy policy.....	152
Trademarks.....	152

# 1 Adult Social Care accelerator

---

Adult Social Care is a simple social program application that is built on top of the Cúram Platform. The program delivers a defined monetary benefit to clients deemed eligible for the Adult Social Care program. The program represents a basic path through a social program application.

## 1.1 Overview of Adult Social Care accelerator

---

The Adult Social Care accelerator provides a straightforward, worked example of a basic application work flow. The example is designed to familiarize you with the key features and underlying infrastructure of a basic application.

### The Adult Social Care business scenario

The goal of the Adult Social Care scenario is to provide benefit payments to incapacitated individuals.

A typical Adult Social Care scenario can address the needs of John, who is aged 48. John is unable to work due to a car accident. Based on a doctor's diagnosis, and the receipt of a practitioner certificate, John receives Sickness Benefit payments. John continues to receive Sickness Benefit payments while his doctor deems he is unable to work, and issues a practitioner certificate that states the underlying diagnoses. If John receives Sickness Benefit payments for more than six months, he automatically moves on to Invalidity Benefit. Invalidity Benefit is paid at a higher rate than Sickness Benefit.

The Adult Social Care workflow implements a scenario in which a case owner is assigned a task to extend a practitioner's certificate that is about to expire. When a payment is generated on a product delivery case, the application assigns a task to the case owner if the certificate is within a pre-configured period of expiration. The case owner must then modify the certificate from the evidence workspace and save the changes.

### Adult Social Care example requirements

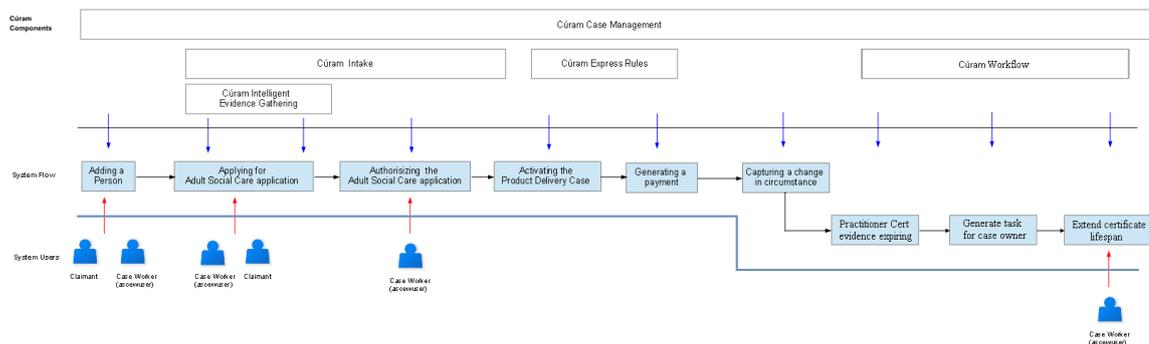
In the example, the team that is deploying Adult Social Care have the following caseworker requirements.

- Search for existing clients or register new clients in the system.
- Create an application for a client.
- Manage case evidence.
- Approve an application.
- View the eligibility and entitlement information for a client.
- Generate a benefit payment to the client.
- Extend a practitioner certificate based on an automatically assigned task.

- Capture a change of circumstance and reassess the client’s eligibility and entitlement for benefits.

## The Adult Social Care business process

A client and caseworker interact to gather and submit the client information. The client information is assessed against a set of business rules to determine the client's eligibility for Adult Social Care benefits.



The Cúram Platform provides the core processing and components necessary to enable this work flow. The work flow uses the following components to satisfy the requirements:

- **Cúram Case Management**  
works to identify high-risk, high-need people and intervene with social program, which results in better outcomes for the individual, in this case a financial payment.
- **Cúram Intake**  
provides a cohesive and flexible benefit intake application process.

- **Cúram Intelligent Evidence Gathering**  
provides an interactive questionnaire to dynamically guide the claimant or caseworker through the process of information gathering.
- **Cúram Express Rules**  
is a language for defining questions that can be asked, and the rules for determining the answers to those questions. In this case, an assessment that is used to determine benefit decisions.
- **Cúram Workflow**  
provides a framework for the configuration of workflows for the assignment of applications and related tasks to workers. Workflows can be configured to reflect the workflow process that an organization requires.

## Adult Social Care design decisions

To meet the Adult Social Care requirements, the team must make key design decisions in the following product areas. The design decisions are documented in detail in the Analyzing the Adult Social Care application section.

- Initial system configuration
- Organization structure
- User application views
- Intake process
- Case management
- Evidence types
- Case ownership strategy
- Eligibility determination
- Financial processes

The accelerator also provides information to understand each of the tasks that are involved in building a basic application. This information is provided in the Building the Adult Social Care application section.

### Related concepts

[Analyzing the Adult Social Care application on page 25](#)

Use the following practices for analyzing the default functionality against the customer's business processes and to document project requirements.

## 1.2 Installing the Adult Social Care accelerator

---

The Adult Social Care accelerator is available for download only to members of Merative such as Lab Services developers who are assigned to work on a Cúram project on a customer site. However, you can reproduce the sample application on the default product by following the detailed step-by-step instructions in Building the Adult Social Care application.

## Prerequisites

You must follow the prerequisites before you install the accelerator.

### Procedure

1. Install a Cúram Application Development Environment (ADE) for application development. The basic development environment consists of the Cúram Application Development Environment (ADE), a Java IDE, and IBM® Rational® Software Architect Designer.
2. Install Cúram Platform.

## Installing and building the accelerator

After you download the accelerator, complete the following steps to install and build the accelerator.

### Procedure

1. Extract the Adult Social Care accelerator at the root of the installation `%CURAM_DIR%`. `%CURAM_DIR%` is the Cúram installation directory, which by default is `C:\Merative\Curam\Development`.
2. From the `%CURAM_DIR%` directory, update the `SetEnvironment.bat` and `SetEnvironment.sh` files.
3. Edit the `SERVER_COMPONENT_ORDER` to add the following components after the custom component.

```
IncapacityBenefit,AdultSocialCare,ASCIntakeConfiguration
```

4. Edit the `CLIENT_COMPONENT_ORDER` to add the following components after the custom component.

```
AdultSocialCare,IncapacityBenefit
```

5. Edit the `%CURAM_DIR%\EJBServer\project\config\deployment_packaging.xml` file to update the Cúram EAR components element to add the following components after the custom component.

```
AdultSocialCare,IncapacityBenefit
```

6. From your Cúram server development installation directory `%CURAM_DIR%\EJBServer`, run the following commands.

```
build clean server database
```

```
build runExtractor createClasspaths
```

7. From your Cúram web client development installation directory `%CURAM_DIR%\webclient`, run the following command.

```
build clean client
```

- Refresh the workspace within the Cúram Application Development Environment (ADE) and start the application as normal.

For information about how to install a development environment, see the *Development Environment Installation Guide*.

## Starting the XML server

Before you start the Cúram application, you must start the XML server in your ADE.

### Procedure

- Change to the `%CURAM_DIR%\CuramSDEJ\xmlserver` directory.  
`%CURAM_DIR%` is the Cúram installation directory, which by default is Windows `C:\Merative\Curam\Development` or Linux `opt/merative/curam/development`.
- Run the following build target to start the XML server:

```
ant -f xmlserver.xml
```

## 1.3 Performing the Adult Social Care case worker scenario

The Adult Social Care caseworker scenario captures data about the client when they apply to an organization for invalidity benefit. It works through various tasks that must be completed to pay a client for the benefit.

### Before you begin

You must have the XML server running to perform the Adult Social Care caseworker scenario.

## Adding a person

Add a person by searching for an existing person or registering a person. If the person does not exist, you can register the person.

### Searching for an existing person

Search for an existing person.

### Procedure

- Log in to the application with the `ascewuser` user credentials.
- From **Quick Links**, select **Search for a Person** link.
- Enter the following values on the **Person Search** tab and select **Search**.

Name	Description
First Name	Given name as stated on official document.
Last Name	Surname as stated on official document.

4. From the **Search Results** window, select the **Person** link for any returned results, to open the Person's home page.

### **Registering a person**

Add a person by registering a new person.

#### **Procedure**

1. Log in to the application with the ascewuser user credentials.
2. From **Quick Links**, select **Register a Person** link and click **Next**.
3. Enter the following mandatory details and click **Save**. A tab opens for the newly registered person.

<b>Name</b>	<b>Description</b>
First Name	Given name as stated on birth certificate.
Last Name	Surname as stated on birth certificate.
Gender	The classification of the Person as either Male or Female.
Date of Birth	An identification date to record the date, month, and year of birth.
Registration Date	The date on which the Person was registered.
Marital Status	The code to denote the state of marriage of the Person.
Nationality	The code to denote the nationality of the Person.
Country/Region of Birth	The country/place of birth of the Person.
Address	By default, Cúram supports 3 address layout types: <ul style="list-style-type: none"> <li>• US address format.</li> <li>• UK address format.</li> <li>• Free form address format.</li> </ul>

## **Applying for Adult Social Care**

Create an application by entering the client details, a practitioner certificate, and diagnosis information.

### **About this task**

When you apply for Adult Social Care for a client, the application case creation process includes a search and match capability. This feature attempts to match clients on a new application to registered persons on the system based on configured search criteria. A prospect person is created on the system if there was no conclusive match.

The person match search criteria that is configured for Adult Social Care include the following:

Table 1: Person match search criteria

Name	Description	Weight
<b>Alternate ID List</b>	The person search matching occurs across all identifiers that are supplied to it, not just the primary alternate identifier. The match now happens on the identifier value and type.	40
<b>Date of Birth</b>	The date of birth for the client.	30
<b>First Name</b>	The given name for the client.	15
<b>Last Name</b>	The surname for the client.	15
<b>Reference</b>	This is the primary alternate identifier for the client. For the Adult Social Care application form, this is the SSN number.	40

## Procedure

1. On the person home page, select the **Care and Protection** tab and select **Applications** from the side navigation bar.
2. From the action menu, click **New Application Form**.
3. From the **Information About The Claimant** window, enter the following values and click **Next**.

Name	Description
Application Date	The date and time on which the application was submitted.
Method of Receipt	The method by which the application was received by the organization. For example, Online or Paper.
First Name	Given name as stated on birth certificate.
Last Name	Surname as stated on birth certificate.
Address	By default, Cúram supports these address layout types: <ul style="list-style-type: none"> <li>• US address format.</li> <li>• UK address format.</li> <li>• Free form address format.</li> </ul> <div style="border: 1px solid blue; padding: 5px; margin-top: 10px;"> <p><b>Note:</b> You need to enter a valid address and zip. See the Adult Social Care example values.</p> </div>
Email Address	Email address for the Person.
What is the claimant's preferred communication Language?	The preferred language for claimant on the application.
Does the claimant require an interpreter?	Indicates whether an interpreter is needed for the interview.

4. From the **Claimant Details** window, enter the following values and click **Next**.

Name	Description
Does the claimant have an SSN?	The code for the Person has a Social Security Number.
Social Security Number (SSN):	A primary identification number, SSN in the US.
Date of birth	An identification date to record the date, month, and year of birth.
Gender	The classification of the Person as either Male or Female.
Marital status	The code for the marital status.
Is the claimant a US Citizen or US Non Citizen National?	The code for the Person is a US Citizen.
When did the claimant become a member of the household?	The date on which the Person became a household member.
Is the claimant a resident of this state?	The code for the Person is residence in the State.
Residency Status	The code for the Residency Status.
What is the claimant's living arrangement?	The code for the Living Arrangement.
What is the status of this living arrangement?	The code for the Living Arrangement Status.
When did this living arrangement begin?	The date on which the Person became a household member.
Race and Ethnicity	The code for the Race and Ethnicity.

- From the **Review The Claimant's Answers** window, ensure that the information is correct and click **Next**.
- From the **Incapacity Benefit Information** window, enter the following value and click **Next**.

Name	Description
Are there examination details for the client?	The code for the examination details is available for the claimant.

- From the **Practitioner Certificate Details** window, enter the following values and click **Next**.

Name	Description
What is the medical certificate type?	The code for the Medical Certificate types.
On what date did examination take place?	The date on which the medical examination took place.
Has the practitioner certified to a specified date?	The date to which the Medical Practitioner certified the Medical Certificate.
What date has the practitioner certified the client fit to return to work?	The date the claimant can return to work.
What is the practitioner's name?	Name of the Medical Practitioner.

Name	Description
What date did the practitioner sign the certificate?	The date the Medical Certificate was signed by the Medical Practitioner.
On what date does certification start?	The date the Medical Certificate begins.
Has the claim form been signed by the client?	The code for the Medical Certificate was signed by the Person.
Are there diagnosis details for the incapacity benefit?	The code for the diagnosis details is available for the Incapacity Benefit program.

8. From the **Diagnosis Details** window, enter the following values and click **Next**.

Name	Description
What ailment is the client suffering from?	The code for the a physical disorder or illness the claimant is suffering.
When was the examination date?	The date on which the medical examination took place.
Are there more ailments diagnosed for client?	The code for more Diagnosis evidence is available.

9. The application questionnaire loops when more diagnoses are available.
10. From the **Review Practitioner Certificate Information** window, review the claimant's practitioner certificate and diagnosis information and click **Next**.
11. From the **Review The Claimant's Answers** window, review the summary of all the claimant's answers and click **Next**.
12. From the **Submit Application Form** window, select **Yes** to submit the application form.
13. Select the **Care and Protection > Applications > Cases** tab to view the application case.
14. Select the reference number to open the application case.

### ***Application case values for Adult Social Care***

Specify the following values when applying for Adult Social Care.

Table 2: Application values

Name	Value
Application Date	Today's date
Method of Receipt	Online
First Name	James
Last Name	Smith
Address	1074, Park Terrace, Fairfield, Midway, Utah
Zip	12345
Email Address	James@curamssoftware.com
What is the claimant's preferred communication language?	English
Does the claimant require an interpreter?	No

Table 3: Application values

Name	Value
Does the claimant have an SSN?	Yes
Social Security Number (SSN):	24684
Date of Birth	9/26/1964
Gender	Male
Marital Status	Married
Is the claimant a US Citizen or US Non Citizen National?	US Citizen
When did the claimant become a member of the household?	9/26/1964
Is the claimant a resident of this state?	Yes
Residency Status	Permanent
What is the claimant's living arrangement?	Home
What is the status of this living arrangement?	Permanent
When did this living arrangement begin?	9/26/1964
Race and Ethnicity	White or Caucasian

Table 4: Application values

Name	Value
Are there examination details for the client?	Yes

Table 5: Application values

Name	Value
What is the medical certificate type?	Doctor
On what date did the examination take place?	Today's date
Has the practitioner certified to a specified date?	Yes
What date has the practitioner certified the client fit to return to work?	Monday, five weeks from now
What is the practitioner's name?	Dr. John Burns
What date did the practitioner sign the certificate?	Today's date
On what date does the certification start?	Today's date
Has the claim form been signed by the client?	Yes
Are there diagnosis details for the incapacity benefit?	Yes

Table 6: Application values

Name	Value
What ailment is the client suffering from?	Neck Pain
When was the examination date?	Today's date
Are there more ailments diagnosed for the client?	Yes

Table 7: Application values

Name	Value
What ailment is the client suffering from?	Lower Back Pain
When was the examination date?	Today's Date
Are there more ailments diagnosed for the client?	No

## Authorizing the Adult Social Care application

Authorizing the application creates an associated integrated case and product delivery case. All evidence is shared from the application to the integrated case.

### Procedure

1. From the **ACTIONS** menu and click **Authorize**.
2. From the **Authorize Program** window, Enter optional comments if needed and click **Save**.
3. Select the refresh icon and ensure the status is set to Closed.
4. From the **Authorization** tab, select the reference number for the newly created integrated case.

## Activating the product delivery case

Activate the product delivery case and viewing the determinations of the eligibility and entitlement results.

### Procedure

1. From the integrated case **Home** tab, select the reference number under the **Cases** cluster to open the product delivery case.
2. From the **Actions** menu, select **Activate Case**.
3. Select the **Determinations** tab.
4. Select the **Current Determinations** menu and click **Decisions** tab.
5. Expand the coverage period. You can see a summary of the eligibility and entitlement results, including a list of practitioner certificates and the diagnoses.

## Generating a payment

Generate a payment to the person on the product delivery case.

### Procedure

1. From the product delivery case, select **Financials > Transactions** tab.
2. Click **Issue Payment** button and click **Yes**.
3. Click the refresh icon to see the payment transaction.

## Performing the Adult Social Care workflow

Performing the Adult Social Care workflow.

### **Triggering the Adult Social Care workflow**

For demonstration purposes, you can implement this workaround to trigger the workflow. You can complete the following tasks to first trigger a workflow task to extend a practitioner certificate, and to renew the certificate.

### **Before you begin**

Before you perform this task, you must complete the previous generating a payment step to generate the first payment on the product delivery case.

### **About this task**

This task describes how to override the server date on your application, enabling you to change your server date to any date and generate the payment for a scheduled date.

### **Procedure**

1. From the `%CURAM_DIR%\EJBServer\project\properties` directory, open your `bootstrap.properties` file in any text editor.  
`%CURAM_DIR%` is the Cúram installation directory, which by default is `C:\Merative\Curam\Development`.
2. Add the following line to the top of your file and substitute in your own date. The specified date for this workflow is the Monday before your practitioner's certificate's end date, which you created as part of performing the Adult Social Care scenario:

```
curam.test.override.date=YYYYMMDD
```

3. Save your changes.
4. From the Eclipse workspace, shut down the Cúram application.
5. From Eclipse project explorer perspective, right-click on your `EJBServer` project and click **Refresh**.
6. Start the Cúram application.
7. From the Cúram application, open your existing product delivery case.
8. Generate a payment for your case.

### **Performing the Adult Social Care workflow task**

Perform the following steps to complete the case owner's task for the Adult Social Care workflow.

### **Before you begin**

Before you perform this task you must complete the payment generation step for the workflow, which results in the creation of the case owner's task.

### **Procedure**

1. From the Cúram application, click **Inbox > Shortcuts > Available Tasks > Search**
2. Click the **Practitioner certificate needs to be extended** task.

3. From the task workspace, click **Actions > Add to My Tasks > Save** to take ownership of the task.
4. From the **My Tasks** tab, click your task.
5. From the **Primary Action** inside the task workspace, click **Evidence Workspace Home Page** to go to the evidence workspace.
6. From the evidence workspace, click the button to the left of the **Practitioner Certificate** evidence to open it.
7. To the right of your open evidence, click **Action > Edit**.
8. From the **Edit Practitioner Certificate** window, modify the **Specified Certification to date** value to extend the certificate.
9. Click **Save**.
10. From the **Evidence Workspace Home Page**, click **Actions > Apply Changes** to complete the evidence modification and close the task.

## Capturing changes in circumstance

There are different ways to capture a change in circumstance. This section covers the different ways in which that can happen in Adult Social Care, as well as outlining the effect it has on determinations and payments.

### ***Changing the Incapacity Benefit Product Type***

Change the evidence to show how it affects case determinations and payments.

#### **About this task**

Changing the Incapacity Benefit Product Type evidence from Sickness Benefit to Invalidity Benefit issues a higher payment amount due to the invalidity benefit having higher rates. The change in benefit also results in changes to the case determinations.

#### **Procedure**

1. From the integrated case **Evidence** tab, select **Active Evidence** menu.
2. Expand the **Incapacity Benefit Product Type** evidence type and select **Edit**.
3. Change **Claim Type** from **Sickness Benefit** to **Invalidity Benefit** and click **Save**.
4. Click **Apply Changes**.
5. From the **Apply Changes** window, select all the evidence types check box to apply changes and click **Save**.
6. From the product delivery case, select the **Determinations** tab to see the amount has changed for the eligible decision.
7. Select the **Financials > Transactions** tab.
8. Select **Issue Payment** and click **Yes**.
9. Click the refresh icon to see the payment transaction.
10. Select the **Over and Under Payments** menu to view the over and under payment records.

### ***Adding a practitioner certificate with new diagnoses***

Add a practitioner certificate with new diagnoses to bring the claimant beyond the six-month mark on benefits. This practitioner certificate reflects a visit to a specialist to highlight the fact that the claimant has serious health issues and is incapacitated for a lengthy period. This new practitioner certificate results in the claimant moving from Sickness Benefit to Incapacity Benefit. The new practitioner certificate and diagnoses can be added by using either the dynamic evidence pages or the Evidence Management wizard. Both alternatives are outlined here.

### **Adding evidence by using the dynamic evidence creation screens**

Add a practitioner certificate with new diagnoses by using the dynamic evidence screens. This practitioner certificate results in the claimant who is receiving Sickness Benefit moving to Incapacity Benefit after six months.

### **Procedure**

1. From the integrated case **Evidence** tab, select the **In-Edit Evidence** menu.
2. Select the page level menu and choose the **New Evidence** option.
3. From the evidence type list page, select the row level action to **Add** a new practitioner certificate.
4. In the **New Practitioner Certificate** window, specify the following details.

<b>Name</b>	<b>Description</b>
Received Date	The received date of the practitioner certificate evidence.
Claimant	The claimant who is receiving benefits.
Medical Practitioner	Specify Service Supplier in the drop-down and use the magnifying glass to open the pop-up search page. Search for Tony Taylor and use the Select link to choose the record returned.
Certificate Type	The type of certificate, Doctor, Dentist, Physiotherapist, and so on.
Certificate From Date	The date from which the practitioner certificate is valid. This date is one day after the specified to date of the previous practitioner certificate.
Specified To Date	The date the practitioner certificate is valid to. This date is six months later that the certificate from date.
Examination Date	The date on which the examination took place.
Signature Date	The date on which the practitioner certificate was signed.
Claim Form Signed	Flag to indicate whether the certificate was signed by the practitioner.

5. Select the page level menu and choose the **New Evidence** option.
6. From the evidence type list page, select the row level action to **Add** a new diagnosis.
7. In the **New Diagnosis** window, specify the following details.

<b>Name</b>	<b>Description</b>
Received Date	The received date of the diagnosis evidence.
Diagnosis	The diagnosis specified by the specialist.
Examination Date	The date on which the examination took place.

8. Repeat steps 5, 6 and 7 to add more diagnoses.

9. Select the page level menu and choose the **Apply Changes** option.
10. In the **Apply Changes** window, select both evidence types and click **Save**.
11. On the Incapacity Benefit product, select the **Determinations** tab and examine the decisions.

A change in benefits is reflected as a result of the new practitioner certificate and diagnoses being added.

*Practitioner certificate and diagnosis values for a change in circumstance by using the evidence screens*

Specify the following values for the practitioner certificate and diagnosis for a change in circumstance by using the evidence screens in Adult Social Care.

Table 8: Practitioner Certificate values

Name	Description
Received Date	Today's date
Claimant	James Smith
Medical Practitioner	Tony Taylor
Certificate Type	Doctor
Certificate From Date	One day later than the last practitioner certificate to date
Specified To Date	Certificate From Date + 26 weeks
Examination Date	Today's date
Signature Date	Today's date
Claim Form Signed	Yes

Table 9: Diagnosis values

Name	Description
Received Date	Today's date
Diagnosis	Paralysis
Examination Date	Today's date

### **Adding evidence by using the Evidence Management wizard**

Add a practitioner certificate with new diagnoses by using the Evidence Management wizard. This practitioner certificate results in the claimant who is receiving Sickness Benefit moving to Incapacity Benefit after six months.

### **Procedure**

1. From the **Actions** menu on the Adult Social Care integrated case, select **Guided Change > Add Medical Condition** to start the Evidence Management wizard.
2. In the first page of the Evidence Management wizard, specify the following practitioner certificate details.

Name	Description
Date Reported	The date on which the evidence change was reported.
Claimant	The claimant who is receiving benefits.

Name	Description
Medical Practitioner	The name of the practitioner who issues the certificate. Use the magnifying glass to search for a Service Supplier stored on the system. Search for Tony Taylor and use the Select link to choose the record returned.
Certificate Type	The type of certificate, Doctor, Dentist, Physiotherapist, and so on.
Certificate From Date	The date from which the practitioner certificate is valid. This date is one day after the specified to date of the previous practitioner certificate.
Specified Certificate To Date	The date the practitioner certificate is valid to. This date is six months later that the certificate from date.
Examination Date	The date on which the examination by the practitioner took place.
Signature Date	The date on which the practitioner certificate was signed by the practitioner.
Claim Form Signed	Indicates whether the certificate was signed by the practitioner.

3. Select **Next** to move to the next screen in the Evidence Management wizard to capture the diagnoses.
4. Specify the following diagnoses details.

Name	Description
Diagnosis	The diagnosis as specified by the practitioner.
Examination Date	The date on which the examination by the practitioner took place.

**Note:** The diagnosis page of the Evidence Management wizard allows for multiple diagnoses to be added before you progress to the next page.

5. Click **Add New Diagnosis** to add the diagnosis details to the **New Diagnosis Details** list.
6. Click **Next** on the **Diagnosis Details** page to progress to the **Summary** page.  
The **Summary** page summarizes the practitioner certificate and diagnoses details added in the first two pages of the Evidence Management wizard.
7. Click **Finish** to complete the Evidence Management wizard and navigate automatically to the **Guided Change** page.  
An entry for **Add Medical Condition** is displayed in the list.
8. Select the row level action **Complete** to complete the change in circumstance.  
If no diagnoses details were entered, the **Guided Change** page would prompt the user to add the diagnoses first by an entry in the **Action Required** list.
9. Select the page level menu and choose the **Apply Changes** option.
10. In the **Apply Changes** window, select both evidence types and click **Save**.
11. On the Incapacity Benefit product, select the **Determinations** tab and examine the decisions.  
A change in benefits is reflected as a result of the new practitioner certificate and diagnoses being added.

*Practitioner certificate and diagnosis values for a change in circumstance by using the Evidence Management wizard*

Specify the following values for the practitioner certificate and diagnosis for a change in circumstance by using the Evidence Management wizard in Adult Social Care.

Table 10: Practitioner Certificate values

Name	Description
Date Reported	Today's date
Claimant	James Smith
Medical Practitioner	Tony Taylor
Certificate Type	Doctor
Certificate From Date	One day later than the last practitioner certificate to date
Specified To Date	Certificate From Date + 26 weeks
Examination Date	Today's date
Signature Date	Today's date
Claim Form Signed	Yes

Table 11: Diagnosis values

Name	Description
Received Date	Today's date
Diagnosis	Paralysis
Examination Date	Today's date

### 1.4 Analyzing the Adult Social Care application

Use the following practices for analyzing the default functionality against the customer's business processes and to document project requirements.

#### Completing a fit gap analysis

You can use the following guidance to complete a fit gap analysis between the project requirements and the default application.

**Fit gap analysis scenario**

Fiona, a business analyst, is asked to do a fit gap analysis of the differences between the Adult Social Care application and the customer’s project requirements. Fiona is focusing on the requirements for the practitioner certificate and person phone details.

The customer's project requirements are:

- The system must capture practitioner certificate details including the client name, medical certificate type, exam date, certification to date, certified return to work date, practitioner name, date certificate was signed, certification start date, inpatient indicator, and facility name.
- The system must capture the client's home telephone number.

At the start of the fit gap analysis, Fiona must have a solid understanding of the Cúram default features. She has read the Performing the Adult Social Care caseworker scenario documentation.

### **Related concepts**

[The Adult Social Care business process on page 10](#)

A client and caseworker interact to gather and submit the client information. The client information is assessed against a set of business rules to determine the client's eligibility for Adult Social Care benefits.

### **Analyzing the default functionality**

You can use the following resources to help you with your analysis.

#### **Reading product information**

You can read the product documentation to learn about Cúram product features.

The product documentation includes business and technical documentation for specific versions and editions of the Cúram product family. Links to Redbooks and white papers are also provided when articles about the product are available. Business analysis documentation is written to provide an overview of the business processes supported by Cúram applications. You can use this information to understand the default features, and how they can be applied to your business processes.

For the business analyst scenario, Fiona finds documentation on how phone numbers are managed in Cúram. The documentation provides a high-level overview of how phone numbers are maintained as person evidence.

For more information about phone numbers, see the *Participant Guide*.

### **Evaluating the default application scenario**

Fiona logs in to the Cúram application to explore the default application features to see whether they meet the project requirements, or if there are gaps. Fiona is analyzing the practitioner certificate and person phone project requirements.

#### **Complete the Adult Social Care application intake script**

Fiona completes the Adult Social Care application intake script to see if the script captures the required data.

Fiona creates an application case by entering the client's details, a practitioner certificate, and diagnosis information. After examining the default Adult Social Care application, Fiona notes gaps in the script. The practitioner certificate questions do not capture the inpatient indicator and facility name.

### **Related tasks**

[Applying for Adult Social Care on page 14](#)

Create an application by entering the client details, a practitioner certificate, and diagnosis information.

### **Examining the practitioner certificate evidence**

Fiona examines the default practitioner certificate evidence to see whether there are any gaps.

### **Procedure**

1. Select the **Care and Protection > Applications > Cases** tab to view the application case.
2. Select the reference number to open the application case.

3. Select the **Evidence** tab to view the default evidence types that are associated with the application case. Fiona can see that the inpatient indicator and facility name are not captured on the practitioner certificate evidence. This is a gap.

### Examining person evidence

Fiona is analyzing the person phone requirements. After reading participant documentation, she knows there is a fit with the Cúram product, but needs to do further analysis to determine the alignment. She decides the next thing she should do is explore the person phone evidence on her local installation of the Adult Social Care application.

### Before you begin

The person must exist on the Cúram system.

### Procedure

1. From the **Person** tab, select the **Evidence** tab and select **New Evidence** from the page level menu.
2. The system displays the **New Phone Number** window. Select the **Phone Type** drop-down field.

### What to do next

Fiona can see the phone number details on the New Phone Number modal page. She verifies it is the same as the layout required for her project. She sees that the label 'Phone Number' is used instead of 'Telephone', which is inconsistent with other systems in her project. Fiona notes this as a gap. She also notes that 'Home' is not displayed in the Phone Type code table.

Now that Fiona has tested the application's person phone functionality, she knows that there is a medium-level solution fit since configuration is required.

### *Cúram Analysis Documentation Tooling (CADT)*

The Cúram Analysis Documentation Tooling (CADT) is a tool that technical users can download and run on a Cúram application to generate analysis documentation that is specific to that application. Technical users can share the analysis documentation with non-technical users to facilitate technical and business analysis of a Cúram application. The analysis documentation primarily supports fit gap analysis and customization impact analysis that relates to Cúram pages and database tables.

Use the analysis documentation to help you to explore the metadata and business functions of an Cúram application and the interactions between them. It supports you in exploring the metadata that is associated with those pages and database tables. In addition, you can see information about related artifact types like Tabs, Domain Definitions, Code Tables, Message Files, Application Properties, and Workflows.

The Cúram Analysis Documentation Tooling (CADT) is available for download from [Cúram Support](#). You must request access to download software.

### *Analyzing requirements using the Cúram Analysis Documentation Tooling (CADT)*

You can use the Cúram Analysis Documentation to determine which application artifacts are used by a particular feature, in this case the person phone feature. Fiona can further explore application

artifacts by using the Cúram Analysis Documentation, which is generated by using the Cúram Analysis Documentation Tooling (CADT).

### Before you begin

The Cúram Analysis Documentation Tooling (CADT) is run on the Adult Social Care application that is built on top of the Cúram Platform and Fiona is provided with the Cúram Analysis Documentation.

### About this task

You can complete these steps of the design process, but information is also provided on how Fiona completes them as part of the Fit Gap Analysis scenario.

### Procedure

1. From the Cúram Analysis Documentation search text box, enter **Phone**, select **Pages** from the drop-down list, and click **Search**.

Fiona can see from the search results several pages that contain the string 'phone'. Fiona wants to understand the context in which phone numbers are used in Cúram. By glancing at the pages that contain the string 'phone', she sees that phone number details are used in the Cúram application pages.

2. From the search results, select **New Phone Number (Participant\_createPhoneNumber)** from the Name column. You see the **New Phone Number** page details.

Fiona verifies that it is the same as the layout used for her project. She sees that the label 'Phone' is used instead of 'Telephone', which is inconsistent with other systems in her project. Fiona notes this different label as a gap.

3. Scroll to the Fields section. Select the **Phone\_Type\_Code** from the Fields section. You see the Domain Definition details for **Phone\_Type\_Code**.
4. From the **Options** domain definition details, select the **PhoneType** link to see the code table details for the phone type code table.

Fiona observes that there is no 'home' telephone number type in the PhoneType code table. This is a requirement for her project, so she notes this as a gap; however, she needs to determine if this is a requirement across the organization.

5. From the **Related Pages** section, it contains a list of pages that use the PhoneType code table. Any modification made to the PhoneType code table impacts all of the pages in this list.
6. Scroll to the Phone section. You see the phone number attributes.

Fiona verifies that the phone number consists of the same three elements as her project's phone number (country code, area code, and phone number). She confirms that the number can be stored in the format that is required by the project.

7. Scroll to the Messages section. You see the validations for telephone numbers and an indicator to show if it is configurable. You see the fields that are mandatory and the ones that are conditionally displayed.

Fiona verifies that the messages are consistent with the project's requirements.

8. From the Cúram Analysis Documentation search text box, enter **Phone** and select **Code Tables** from the drop-down list and click **Search**.

Fiona can see from the search results a list of code tables that contain the string 'phone'. One of these code tables is the BusinessPhoneType code table. Fiona remembers when she was assessing the impact of changing the word 'phone number' to 'telephone', there were other screens in the application that are related to phone numbers that the project plans to use, but that don't seem to use the PhoneType code table. She wants to make sure that there are no changes that are required to any other code tables. Based on the information that is shown to her for the BusinessPhoneType code table, she realizes that this code table contains phone number types that businesses would use and should therefore not contain the value 'home'. This is not a gap.

## Analyzing the organization structure

You must map your organization and locations to the appropriate Cúram artifacts to represent your installation.

### ***Cúram organization structure***

The organization structure provides a logical representation within the application of how the organization is physically structured. The organization structure consists of organization units, positions, and jobs and their relationships to each other, and the users of the system and where those users fit into the overall structure.

The organization structure provides a reporting hierarchy that is used throughout the application to determine how case management activities, such as supervisory approval tasks, are managed across the organization.

For more information about the organization structure, see the *Organization Administration Guide*.

### ***Defining the organization structure requirements***

You must define the organization structure requirements for your organization.

### ***Defining the organization unit requirements***

You must define the organization units in your organization structure.

### **About this task**

You must construct the organization units in levels to represent the various departments within the organization and how they report to each other, taking into consideration whether or not you have regional administration. For example, an organization unit that is created to represent one regional department within the organization might have five child organization units that are created under it to represent five divisional departments that report into the regional department.

## **Defining the job requirements**

You must define the job requirements for your organization structure.

### **About this task**

Cúram allows you to manage work distribution for the organization according to job and user skills. Depending on your requirements you might need generic jobs (such as ‘worker’ or ‘supervisor’) or more granular jobs to allocate work to a particular job.

You must define the jobs in your organization and then associate each job to one or many positions. You can then use the job and its associated positions for task allocation. For example, if you create 1 job and 10 positions, and assign 10 users to each of the 10 positions, you can allocate the task in question to the job such that all users would see it in their inbox.

## **Defining position requirements**

You must define the positions for each organization unit, and the reporting requirements between the positions.

### **About this task**

The positions should be constructed to build out an appropriate reporting hierarchy between users within the organization, both within and across organization units. By default, the position (reporting) hierarchy is inherited through the organization unit hierarchy and the configuration of lead positions in the organization units. Optionally, the reporting hierarchy can be further configured within and across organization units by using the 'reports to' assignment for positions. This is important when there are cross organization unit reporting requirements or multiple lead positions or both configured within a single organization unit.

Given the flexibility of reporting positions, there are different options for implementing cross-functional reporting structures. The easiest way to ensure users are reporting to the most appropriate supervisors is to manually set up a reporting hierarchy. For example, in an organization unit consisting of different levels of social workers, it might make sense to set up reporting relationships based on experience. The less experienced social workers could report to more experienced social workers. The more experienced social workers could report to a lead position in a middle-management organization unit.

When you define the 'reports to' association between positions that do not fit into a typical reporting hierarchy such as where one position within an organization unit report to another position within the same organization unit, use the 'Position Search' feature that allows a position to be configured to report to any other position within the organization.

You must consider that if a position has a ‘reports to’ position defined, and more than one user is assigned to the position to which that position reports, the system chooses just one user to assign a task to, rather than all users. The system performs load balancing based on the configurations. For example, if you assign two users to the reports to position and submit a case for approval four times, two of the tasks are assigned to the first user, and two to the second. However, if a position has no ‘reports to’ position that is defined, or there are no users who are assigned to the reports to position, then the system assigns tasks to the user who holds the lead position in the organization unit. If more than one user holds the lead position, then similar to above the system just picks one user to assign the task to. The same occurs if an organization unit has two lead positions.

It is important to note that you cannot assign a user to two positions simultaneously.

***Defining the lead position requirements***

You must define the requirements for the lead positions in your organization structure.

**About this task**

For the most basic reporting hierarchy to exist on the system, there must be at least one lead position in place for user reporting purposes. There must also be at least one user assigned to this lead position. If there is no lead position assigned to the organization unit, the reporting position is automatically set to a lead position assigned to a parent organization unit.

***Defining multiple lead position requirements***

You must decide whether there are requirements to have multiple lead positions in the organization structure.

**About this task**

When multiple lead positions are defined within an organization unit, the default task escalation strategy assigns tasks to users in a lead position by determining which of the users in the multiple lead positions have the least number of tasks that are currently assigned to them. This behavior must be considered when you determine whether an organization unit should have more than one lead position.

The requirement to have multiple lead positions in an organization unit is influenced by how the business operates with regards to supervisor subordinate relationships. Often supervisors share supervision of multiple users. In this instance, the supervisor workspace needs to be considered. The supervisor workspace groups subordinate users according to who they report to. If multiple supervisors need to have the same list of users in their supervisor workspace, those users would need to be all in the same organization unit, or the supervisors must be placed in a parent organization unit to all the subordinate users' organization units.

***Cúram location structure***

The location structure provides a logical representation in the Cúram application of the physical places where an organization's users work and allows an organization to establish the parent-child relationships that exist between locations. The location structure is used to manage security at a location level and control user access to case and client information.

Locations can also be public offices where the public can interact with the organization. A location must be indicated as a public office to identify that location as the location of a participant registered in the application.

In Cúram, each user is assigned a location and a user's access to case and client information is determined based on the location of the user.

***Defining the location structure requirements***

Once you define the structure, you must define the corresponding location structure requirements.

**About this task**

A user's access to case information is determined based on a comparison of the location of the user against either the location of the case owner or the location of the primary client on the case. When you begin defining the location structure requirements for your organization, you must

first define the location structure. You should consider starting from the basis of one location per organization unit and build out from there. You must also consider what access users in one location should have to case and client information that is owned by users in other locations.

### **Defining the location hierarchy requirements**

You must define the root, parent, and child locations for your location structure. Once this is defined it is used to implement Location Based Security.

### **About this task**

The completed Organization Structure is used as the starting point for the Location Structure. There can be only one active location structure at a time.

Each location structure must have a root location. Once the root location is defined, you must define the location hierarchy that includes parent and child locations. The locations that are created for your location structure should be constructed in hierarchical levels to represent how access to case and client information should be secured between the various work places within the organization. For example, if users within a regional location should have access to case and client information across all counties within the region, but the users in one county location should only have access to case and client information that is managed by users within that county, the county location should be created as a child location of the region location.

The Location Structure is defined so that you can implement Location Based Security (LBS), which must be set to one of these settings:

- Off (No restrictions on any access)
- On (Users can only update cases in their own location)
- Restricted View (Users can see that a case or client exists in another location, but they have no access to it)
- Read Only (Users can update cases in their own location and view cases in other locations)

LBS compares the location of the user against the location of case and client information to determine the access rights of the user. The location of the client is derived from the client's preferred public office that is captured on the participant record. The location of the user is determined in the following two ways (if either are successfully compared to the location of the client, then the user is granted access to case and participant information):

- The location that is specified on the user record.
- The location that is specified on the user's position.

Make sure that you capture any additional requirements that are associated with the LBS setting. For example, if an organization requires the LBS setting “Restricted View” for most cases, but for cases in one particular location require the name and address to be hidden to users in all other locations, then customization would be required to hide the name and address fields in all the search list screens.

### ***Defining the security requirements***

You must define the organizational, location, and user security requirements to deploy an installation of Cúram.

### **Related tasks**

[Defining the security requirements on page 32](#)

You must define the organizational, location, and user security requirements to deploy an installation of Cúram.

### **Defining the organizational security requirements**

You must define the organizational security requirements.

#### **About this task**

Each organization unit may have Read, Maintain, and Create Unit SIDs assigned to it. The Create Unit SID is the SID that a user must have to create a child organization unit for the organization unit. The Read SID is the SID that a user must have to view organization unit details and details of its child units and positions. The Maintain SID is the SID a user must have to maintain details of the organization unit and its child units and positions. The Maintain SID and the Create Unit SID always take precedence over the Read SID. i.e. where the user has the right to maintain a unit or create a child unit for a unit, they automatically have the right to view the unit regardless of whether they have read rights or not.

### **Defining the location security requirements**

You must define the location security requirements.

#### **About this task**

Locations can be secured to limit user access to case and client information based on a comparison of a user's location, the location of their positions or positions, and the location of the case and client information.

Locations can also be secured so that only specific users can access location administration details. Security identifiers are used to secure administrative details for a location structure at various levels, such as at a regional level.

Each location has Create, Read, and Maintain SIDs assigned to it. The Create Location SID is the SID that a user must have to create a child location for the location. The Read SID is the SID that a user must have to view administration data within this location and its sub locations. The Maintain SID is the SID a user must have to maintain administration data within this location and its sub locations. Users with maintain SIDs for a location can view administration data for locations.

### **Defining the user security requirements**

As part of any deployment of Cúram, you must define the user data that is required for access to the application.

#### **About this task**

Once the Organization Structure is complete, you can use the Positions as the basis for setting up the User Accounts. Consideration should be given to:

- Position and Location
- Security Role
- Application View

To define the requirements for the security roles and application views, start at a high level and determine:

- Who can see person data?
- Who can see payments?
- Who can see services?
- Who can see other participant types?
- Who can see other data, as required?

It is also important to consider who can update the data that is listed above and to identify any areas where there might be special requirements, for example, amending bank accounts.

The security roles and application views can be determined by examining areas of functionality (for example, Intake, Investigation, On-going Cases) and existing requirement documentation. Once the details are documented, the developers match the security identifiers to your requirements and assign them to the appropriate security groups.

When you define the user accounts, it is important to identify the most “high risk” areas of functionality and separate them into individual security groups, for example, Bank Accounts. It is a balancing act to keep the number of security groups to a minimum without making them either too complex and large, or too granular. It is also important to keep the default ADMIN and SYSADMIN users, as they can prove useful in the testing and live environments.

## Analyzing the user requirements

You must analyze the default application user interface to ensure that it meets your requirements.

### ***User application view***

A user application view is a pre-defined view of the application that is specific to a user role. The user application view displays the application home page, personalized pod pages, sections, shortcuts, searches, tabs, and other user interface components that are relevant to the user role.

### ***User interface components***

An application user interface is composed of a number of elements that allow a user to perform functions on the system.

For more information about the user interface, see the *Web Client Reference Manual*.

### ***Users***

A user is somebody who is employed by the organization to complete tasks on the system, such as creating and managing cases.

New users must be registered on the system and assigned a security role and user application view.

### ***Default user application views***

There are a number of default user application views including the administrator and system administrator. From an adult social care perspective, the most important user application view is the eligibility worker view.

#### **Administrator user application view**

The administrator user application view provides access to the administration configurations for the administrator user.

The administrator user application view allows the user to configure the organization structure, location structure, case structure, rules, evidence, intelligent evidence gathering, intake, among other functional areas.

#### **System administrator user application view**

The system administrator user application view provides access to the system administration configurations for the system administrator user.

The system administrator user application view allows the user to configure application properties. The user can also configure case audit selection queries, communication templates, batch processes, security settings, business intelligence reports, target systems, and content management interoperability services.

#### **Eligibility worker user application view**

The eligibility worker user application view was created to allow a user to process applications and cases. With this view, the user can search and add persons to the system, process an application, create and manage cases, determine eligibility, authorize benefits, and handle change in circumstance processing.

The user application view sections, pods, tabs, and shortcut panel are configurable.

### ***Define the user application view requirements***

As part of any deployment of Cúram, you must identify the functional and data requirements for each of your users. Review the existing user application views and define any additional requirements for your users.

The user role and user application views can be determined by examining functional areas (for example, Outcome Management and On-going Cases) and existing requirement documentation. You must understand which features are required by the user role.

When you define the requirements for the user application view, you must consider the following user application view elements:

- Personalized pod pages,
- Sections,
- Tabs
- Shortcuts, and
- Searches.

You must decide which pods must be pre-configured on the personalized pod pages and which ones can be added at the discretion of the user. You must also consider whether any of the pre-defined pods need to be modified and if there is a requirement for a custom pod.

You need to decide which sections can be displayed for the user role and limit the number of sections to under five sections.

It is also important to review the existing tab structure to determine whether any tabs must be removed if the user role does not require access to the information.

You must also decide whether the default shortcut panel menu items need to be modified. Consider removing any links the user role must not have access to. Conversely, the user role might require more links on the shortcut panel to perform their tasks.

The default application allows the user to perform various searches on the system data. You must decide whether all of the existing search pages that are displayed in the user application view are required by the user role. You might need to modify the search pages if the user role must not have access to some of the data that is presented in the search results.

## Analyzing the intake process requirements

You must refine the intake process to best suit the requirements of your organization.

### ***Intake***

Intake is the process of adding clients to the system. It begins with searching for the client in the system and ends with the authorization of the application case.

The intake process begins by searching for or registering the client in the system. The caseworker then creates an internal application for the registered person with an application form. More clients can be added with the application form, if configured, or from the application case when the case is created. The application form is an Intelligent Evidence Gathering (IEG) script that consists of a set of questions. The questions are aimed at capturing client information that is required to authorize the programs for which the client is applying.

When the internal application form is submitted, the application case creation process includes a search and match capability. The search and match process attempts to match clients on a new application to registered persons on the system based on configured search criteria. After all of the clients are either matched or registered, the application case is created. The application case type is determined based on the configuration settings for each program that is applied for. The application case status is set to Open.

Any data captured on the IEG script is mapped to the application case which is created upon submission. The mapped data is stored as 'in-edit evidence'.

Once the application case is created, the caseworker manages the application evidence and outstanding verifications. Once complete, the caseworker can select to authorize the application case. Authorizing the application case creates an associated integrated case and product delivery case if the client is found eligible. When evidence brokering is configured, the evidence is shared from the application case to the integrated case.

### ***Application case***

Cúram Intake provides an application case to manage the information that is related to an application for programs. An application case can contain a number of associated programs.

Evidence can be required to authorize one or more of the programs that are associated with the application case.

The caseworker can manage the following application case information from within the application case:

- Clients
- Programs
- Timers
- Evidence
- Eligibility checks
- Related cases
- Notes
- Meetings
- Attachments
- Communications

The application case can also be configured to include program timers, eligibility checks, Priority, Complexity and Risk (PCR) assessments, and the ability to create an appeal on a program denial.

### ***Define the application case requirements***

You must decide on a few key decision points when you define the application case requirements.

#### **Define the ownership strategy**

You must define who owns the application case when it is created.

You can choose to assign separate ownership strategies to application cases created internally and application cases that are created as a result of an online application.

- For internal and online applications, you can define that the application case is assigned to a specific organization unit, position, user, or work queue.
- For internal applications only, you can specify that the application case is assigned to the current user, for example, the user who created the application case.

#### ***Adult Social Care ownership strategy***

In Adult Social Care, the internal Adult Social Care application is used by the worker to help the client apply for benefits. The submitted applications are assigned to the Eligibility Worker user role.

#### **Define the program authorization strategy**

You must decide at what level you want programs to be authorized. You can choose to authorize at the application level or at the program level.

##### ***Application level authorization strategy***

You can choose to authorize at the application level. If you decide to authorize at the application level, then all the programs on the application case are authorized together when you select the Authorize action on the application case.

If the authorization strategy is set to application level, you must also define the authorization Cúram Express Rules (CER) ruleset. The ruleset must determine the set of programs to be authorized, and the clients that are eligible for each program.

When the application level authorization strategy is selected, you must also decide whether a new or existing integrated case should be used when the program authorization is successful.

The integrated case is used to host any product deliveries that are created as a result of the authorization. If a new integrated case is created, all of the application case clients are added as case participants to the integrated case. If an existing integrated case is used, any additional clients on the application case are added as case participants to the integrated case. Any evidence that is captured on the application case that is also required on the integrated case is copied to the integrated case upon successful authorization.

#### *Program level authorization strategy*

You can choose to authorize the application at the program level when the application case has one or more program types that are associated to it. Separate integrated cases are used to manage the product deliveries that are created as a result of authorization. If you decide to authorize at the program level, then each program must be authorized separately.

An organization might want its workers to authorize programs manually, without the requirement for an authorization ruleset. In this instance, you can use program level authorization. You can choose to create a new integrated case for each program that is authorized, or that an existing integrated case is used, or that the worker selects one or the other. You must also define the type of integrated case that is created as a result of authorizing the program. If a product delivery is required, the product delivery type must be specified for the program, along with the client selection strategy. For example, the strategy for determining which clients on the application case are added to the product delivery.

#### *Adult Social Care authorization strategy*

For Adult Social Care, the authorization strategy is set to the program level. While either setting is appropriate, the program level authorization strategy was chosen for expediency reasons.

#### **Define the eligibility check strategy**

You must decide whether your project requires eligibility check processing on the application case. If enabled, the worker can check eligibility for the programs on an application case.

If you require this feature, you must specify a CER rule set. The rule set outputs the eligibility decision for the programs that are checked on the application case, and, if relevant, the entitlement for that program. If an Eligibility Check rule set is specified, an Eligibility Check Strategy must also be defined.

You must consider that the case determinations are stored as XML blob and use significant space that might slow down your system. You must decide how you plan to address the storage issue when you implement the eligibility check functionality.

#### *Adult Social Care eligibility check strategy*

For expediency reasons, the decision was made not to enable eligibility check processing on the Adult Social Care application case.

#### **Define the workflow requirements**

You must decide whether you need to define workflows as part of application case creation and processing.

You can specify the following workflows:

- **Application Case Created**

A workflow process starts when the application case is created.

- **Withdrawal Request**

A workflow process starts in response in response to an online request by a citizen to withdraw the application for a particular program.

- **Straight Through**

A workflow process handles straight through processing. For example, from creation of the application case to creation of an ongoing case and closure of the application case. If specified, this process is started upon creation of the application case as a result of an online application submission or an internal application form submission. All of the clients on the application case must be successfully matched or registered before the straight through process is started.

- **Ready for Determination**

A workflow process starts when the application case moves to a status of Ready for Determination. This workflow can be specified only if Ready for Determination is enabled for the application case.

### *Adult Social Care workflow process*

The Adult Social Care system includes a workflow process to notify the case owner to extend the claimant's certification period.

As part of the Adult Social Care intake process, the application script includes questions where the caseworker can specify a date for which a claimant can return to work.

A configurable property is implemented which specifies the number of days for which an event needs to be raised to notify the case owner to extend the certification period.

This event enacts the Adult Social Care work flow and creates a task and assigns it to the case owner. The task is displayed in the case owner's inbox where they can extend the claimants certification period. Once the evidence changes are applied, an event is raised to close the task.

### **Define the evidence types**

Evidence can be required to authorize one or more of the programs that are associated with the application case. You must decide which evidence types are required by the application case and associate them to the application case.

When evidence is associated with an application case, it is automatically enabled for evidence brokering. You must define the brokering that is required between the application case and other case types. You must include integrated cases and product deliveries that are created as a result of program authorization. You can also configure brokering of evidence from the application case to the person or prospect person.

### *Adult Social Care dynamic evidence types*

You must decide which dynamic evidence entities are associated with the integrated case. For Adult Social Care, six dynamic evidence types are configured. The Practitioner Certificate, ASC Diagnosis, and Incapacity Benefit Product Type evidence are configured to be associated with the integrated case type. The evidence can be used by any product deliveries that are associated with that integrated case. The system runs the eligibility and entitlement rules against these evidence types to determine eligibility for benefits.

Entity Type	Description
<b>ASC Household Member</b>	This entity contains details of an individual who lives in the household. Personal information is captured for household members through the system registration process. Additional information such as citizen status, veteran status, and ethnic details, can be captured on the household record.
<b>ASC Living Arrangement</b>	This entity describes the living arrangement of the client, for example, if they live at home, are homeless, or live in an institution.
<b>ASC Alien</b>	If the client is an alien, this entity holds information in relation to the alien. Alien types include refugee, asylee, and parolee. The alien must be determined an eligible alien to receive assistance from certain programs.
<b>Practitioner Certificate</b>	The Practitioner Certificate entity is a representation of a certificate that is received by a medical professional. It contains the from and to dates that the certificate covers, the date of examination, the medical practitioner's name, and it is also signed by the practitioner. In evidence terms, the Practitioner Certificate is a parent of the ASC Diagnosis entity.
<b>ASC Diagnosis</b>	The ASC Diagnosis entity contains the diseases or ailments that are suffered by the client. It also contains the date of examination. In evidence terms, the ASC Diagnosis entity is a child of Practitioner Certificate. It's possible for the client to suffer from one or more ailments, in which case one diagnosis record is created for each ailment.
<b>Incapacity Benefit Product Type</b>	The Incapacity Benefit Product Type entity determines whether the client receives Sickness Benefit or Invalidity Benefit. It is defaulted to Sickness Benefit for the first 182 days and then automatically moves to Invalidity Benefit if the client is still receiving the benefit.

### ***Application intake script***

The application intake script contains sections, question pages, questions, and conditional logic to capture information.

For example, the application intake script might include sections for:

- About the Claimant
- About the Household
- Expenses
- Resources
- Finish

Each section contains question pages that are used to capture information about the claimant.

### ***Define the application intake script requirements***

You must decide whether you need to modify the existing application intake script or create a new application intake script.

You must document the requirements for the application intake script. Using the application intake script as reference, document the requirements for any additions or modifications to the script. The developer then uses the requirements to create or modify the IEG files.

**Person search match**

The application case creation process includes a search and match capability. This feature attempts to match clients on a new application to registered persons on the system based on configured search criteria.

The Person Search Match Criteria define the data that the system uses to search for and match a client on an application with registered persons. The criteria are defined and configured within the Participants section of the administration application.

The weight defines how important the criterion is in determining whether the registered person returned by the search matches the client on the application. For example, the organization might define three search criteria, First Name, Last Name, and Date of Birth. First Name might be weighted at 20, Last Name might be weighted at 30, and Date of Birth might be weighted at 50. The weights are used to determine a score for a registered person who is returned by the search.

The score determines whether the person is a conclusive match, an inconclusive match, or not a match, based on the thresholds that are configured in the system administration application.

**Define the person search match requirements**

You must define the search criteria requirements for your organization.

An organization must decide which search criteria to use, and the weights to assign to each criterion.

**Related tasks**

[Modifying person match on page 127](#)

Person match is a process for automatically matching applicants to registered clients. Complete the following tasks to modify the person match criteria.

**Analyzing case management**

You can must define the requirements for the cases that are used to deliver benefits and services to the client.

**Analyzing the integrated case requirements**

You must decide what evidence types are associated with the integrated case. For each evidence type, you must decide how the evidence is updated, how verifications are configured, and how the evidence is brokered between cases and persons. It is also important to define the requirements for the integrated case ownership strategy.

**Integrated case**

The integrated case provides a holistic view of a client and their family's needs. The integrated case also provides the steps that are taken to address these needs through the delivery of products, services, and ongoing case management. This includes a forum in which clients and agencies can interact with each other, as well as a set of business processes for delivering benefits and services and creating and managing outcome plans. The integrated case brings together all the relevant parties that are involved on a case and provides structure to facilitate the collaboration that is needed to identify and address the needs of the clients.

### Defining evidence types

Evidence types that are associated with the integrated case must be defined. You must decide how the evidence is updated, how to configure evidence verifications, and how the evidence is brokered between cases and persons.

#### *Adult Social Care dynamic evidence types*

You must decide which dynamic evidence entities are associated with the integrated case. For Adult Social Care, six dynamic evidence types are configured. The Practitioner Certificate, ASC Diagnosis, and Incapacity Benefit Product Type evidence are configured to be associated with the integrated case type. The evidence can be used by any product deliveries that are associated with that integrated case. The system runs the eligibility and entitlement rules against these evidence types to determine eligibility for benefits.

Entity Type	Description
<b>ASC Household Member</b>	This entity contains details of an individual who lives in the household. Personal information is captured for household members through the system registration process. Additional information such as citizen status, veteran status, and ethnic details, can be captured on the household record.
<b>ASC Living Arrangement</b>	This entity describes the living arrangement of the client, for example, if they live at home, are homeless, or live in an institution.
<b>ASC Alien</b>	If the client is an alien, this entity holds information in relation to the alien. Alien types include refugee, asylee, and parolee. The alien must be determined an eligible alien to receive assistance from certain programs.
<b>Practitioner Certificate</b>	The Practitioner Certificate entity is a representation of a certificate that is received by a medical professional. It contains the from and to dates that the certificate covers, the date of examination, the medical practitioner's name, and it is also signed by the practitioner. In evidence terms, the Practitioner Certificate is a parent of the ASC Diagnosis entity.
<b>ASC Diagnosis</b>	The ASC Diagnosis entity contains the diseases or ailments that are suffered by the client. It also contains the date of examination. In evidence terms, the ASC Diagnosis entity is a child of Practitioner Certificate. It's possible for the client to suffer from one or more ailments, in which case one diagnosis record is created for each ailment.
<b>Incapacity Benefit Product Type</b>	The Incapacity Benefit Product Type entity determines whether the client receives Sickness Benefit or Invalidity Benefit. It is defaulted to Sickness Benefit for the first 182 days and then automatically moves to Invalidity Benefit if the client is still receiving the benefit.

#### *Evidence updates*

You must decide how evidence is updated. Evidence supports two types of evidence change: evidence correction and evidence succession. A user can modify the effective date of change or else leave it the same as part of an update. The effective date of change is the field, which determines whether a modification to an active evidence record is a succession or a correction.

### Defining evidence correction requirements

You must decide whether evidence can be corrected. An evidence correction is the replacement of an existing evidence record with a new evidence record to correct an incorrect piece of data. It is important to consider this type of update when you define new or modify existing evidence types.

The Effective Date of Change attribute controls whether an update to an evidence record is stored as a correction or a succession. The display or non-display of the Effective Date of Change field is set in the Dynamic Evidence Editor. A correction does not allow the evidence to change over time. The user must end one record before they add a record. The Effective Date of Change field is left blank for a correction to evidence. The evidence type attributes include a From Date and a To Date.

You must consider whether a person can have more than one instance of this information at any point in time. If no, then an evidence correction can be configured. An example of a correction only evidence type is the Address evidence. You can create one or more addresses of the same type. However, the address data does not change over time. When a person moves, they have a new address. The system needs to show change of address over time. The From date and To date are captured on the evidence to represent the period in which the address is in effect.

You might need to add or modify evidence types on your project and consider whether this type of evidence update is required when you are defining the evidence requirements.

#### Defining evidence succession requirements

You must decide whether evidence successions can be used to update evidence. An evidence succession is the set of evidence records of the same type that collectively represent a piece of evidence as it changes over time. It is important to consider this type of update when you define new or modify existing evidence types.

The Effective Date of Change attribute controls whether an update to an evidence record is stored as a correction or a succession. The display or non-display of the Effective Date of Change field is set in the Dynamic Evidence Editor. A succession allows the evidence to change over time and the Effective Date of Change is captured on the evidence. For example, a bank account evidence record can include a bank account balance. This bank account balance is likely to change over time and the succession of bank account balances collectively represent the changes to the bank account.

A succession of changes in circumstance to an evidence record is supported by temporal evidence, so you can see a history of all the changes.

You must decide whether any of the data attributes are likely to change during the lifetime of the evidence record. If yes, then the Effective Date of Change can be configured for this evidence type.

You might need to add or modify evidence types on your project and should consider whether this type of evidence update is required when you are defining the evidence requirements.

#### Adult Social Care evidence configurations

For Adult Social Care, Practitioner Certificate evidence and the ASC Diagnosis evidence are configured to be correction only. The Incapacity Benefit Product Type evidence is configured to allow for evidence succession.

In the Adult Social Care application, the Practitioner Certificate evidence and the ASC Diagnosis evidence are effective for a specified length of time, for example, 12 weeks. As this information does not change over the effective period, a change of circumstance is not allowed and the

Effective Date of Change field is not available when the evidence is being modified. The user must end date one evidence record before a new evidence record can be created. Conversely, the Incapacity Benefit Product Type evidence is configured to allow for evidence successions since it is likely to change over time. In this instance, the effective date of change is available for capture for this evidence as part of an update.

### ***Evidence verification***

You must define the requirements for evidence verification. Client information can be verified by documents such as birth certificates or bank statements. The system identifies whether there are any verification requirements for a particular type of evidence based on how verifications are configured within the administration application. For more information, see the *Evidence Broker Overview Guide*.

#### **Components of verification**

Verification consists of three components: an administration component, a case component, and a participant component.

### **Administration component**

Use the administration component to customize aspects of verification. For example, the following list outlines three ways that the verification component can be customized:

- Customize restricted access to verifiable data.
- Customize specialized processing that is triggered by changes to verified evidence.
- Determine whether a verification is mandatory.

The verification settings can either be applied to one product or can be reused for multiple products. Verification settings can also be applied to an application case and participant evidence.

### **Case and participant components**

#### **• Record verifications for evidence**

Use the case and participant components of the verification process so that caseworkers can record verifications for evidence. To implement the process, the verification engine interprets the rules that were defined during verification administration. The rules include how to identify whether there are any verification requirements for a selected piece of evidence. During the maintenance of the evidence, the verification engine ensures that any rules that relate to the verification are implemented. For example, if two verification items are needed to satisfy a verification requirement then the relevant evidence cannot be activated unless two items are provided.

#### **• Fulfill verification requirements**

Evidence and case list pages are provided to assist caseworkers to fulfill verification requirements. Caseworkers can also view verifications that are related to participant evidence from the participant manager. Caseworkers can use the pages to view either the full list of verifications or the outstanding, that is, unsatisfied, verifications. Caseworkers can also run other functions, for example caseworkers can add attachments, such as graphics files, to verification items.

#### **Defining evidence verification requirements**

You must decide for each evidence type if any of the attributes need to be verified.

When you analyze your project requirements it is important to consider for each evidence type if any of the attributes need to be verified. If yes, there are a number of verification elements to consider when you analyze evidence verifications. Key items include verifiable data items, verification item utilizations, verification requirements, verification requirement usages, and dependent data items. This example uses the Income evidence entity to provide more information on each of these elements.

Name	Description
<b>Verifiable Data Item</b>	The <i>verifiable data item</i> is a piece of evidence that requires verification. This piece of evidence corresponds to a single attribute within a specific evidence entity, for example, an income amount on the income entity. To create a verifiable data item, two attributes must be entered for the evidence entity; the name of the entity (this is stored in the "evidence type" field for an entity) and the exact name of the desired attribute to be verified.
<b>Verification Requirement Usage</b>	The verification engine allows an individual verification requirement to be used by many different types of cases. A <i>verification requirement usage</i> allows administrators to associate specific case types with specific verification requirements. In practical terms, this enables an administrator to specify different evidence verification requirements for different types of cases. For example, a client's income is captured at the integrated case level. If there is a requirement to verify the income, this requirement can be used by multiple cases within the integrated case. Verification requirement usages are beneficial because they allow verification rules to be applied to groups of cases (for example, all the cases within an integrated case), or separately applied to individual cases.
<b>Verification Items</b>	A <i>verification item</i> defines what can be used to verify the information provided by a participant. A pay stub can be used to verify the income amount provided by the participant.
<b>Verification Item Utilizations</b>	<i>Verification item utilizations</i> define which <i>verification items</i> (for example, pay stub) are to be used for a particular verifiable data item. A verification item defines what can be used to verify the information provided by a participant, for example a pay stub or an income tax return. For some verifiable data items, it may be possible to provide a number of options as to how the data item is verified, in which case such verifiable data items will have a number of verification item utilizations. For example, income amount can be verified by providing a pay stub or an income tax return.
<b>Verification Requirements</b>	A <i>verification requirement</i> provides the rules of verification for a piece of data (verifiable data item). There are many variables included in these rules including where and how the rules apply at runtime. For example whether the verification engine needs to apply the rules to participant level data or to a specific case. This time using date of birth as an example of a verifiable data item, for some organizations the rules may be to verify this piece of data once and therefore verification engine applies the rules within the participant manager. For other organizations the rules may require that date of birth is verified at a program level and therefore the verification engine applies the rules to a specific case.
<b>Dependent Data Items</b>	<i>Dependent data items</i> are specific pieces of evidence that have a direct influence on the verification of a related data item. Although these pieces of evidence do not require verification, it may be important to record them for the verification of a related data item. For example, if your organization wants to verify the reason that a household member was absent from the household, the length of the absence may be an important fact to record for the verification. In this example, the "Absence Reason" is the verifiable data item, and the "To" and "From" dates of the absence are dependent data items. The verification engine treats any change to a dependent data item in the same manner as a change to the verifiable data item.

## Related concepts

### Conditional verifications

Define the conditional verifications as verifications that are based on a set of conditions rather than verifications that are based only on added or modified evidence.

### Verification engine

When evidence is added or modified, the verification engine checks the current specified conditions. However, the verification engine creates an outstanding verification record only when a condition that is defined is met. The verification engine does not create an outstanding verification record each time a verifiable data item is added or modified. The conditions can range from conditions against the value of the verifiable data item to more complex conditions where the values of a set of dependent evidences determine whether verification is required.

### Examples

For example, a verification might be required only when the value of earnings is more than \$200 per week or only where the alternate ID is of type SSN. A set of dependent evidences is another, more complex, example. Eligibility for an income assistance program, for example, that requires verification of a household income evidence type when the income is more than \$1,150 per month. The household income evidence type consists of multiple income evidence types, for example dividends, pension, wages, and salaries. The verification is configured for the income amount of the household income evidence type. However, the verification engine reevaluates whether the household income requires verification when the income of any dependent evidence types, that is, dividends, pension, wages, and salaries, changes.

### Rule classes

The verification engine can permit creating a conditional verification so that the user can associate a rule class. The organization must provide the rule classes that define the conditions for the verifiable data item. To use conditional verifications that suit specific business scenarios, the organization must provide three items: a rule class, a display rule class, and a display UIM.

- **Rule class**  
A mandatory rule class is required that defines the conditions for the verification to trigger for the verifiable data item.
- **Display rule class**  
If required, a rule class that defines how the results of the verification are displayed.
- **Display UIM**  
If required, a UIM page reference to display the results of the conditional verifications in the verifications page.

### Defining conditional verification requirements

You must decide if conditional verifications are required for any of the evidence types. Conditional verifications support more complex rules that are used to determine when verifications should apply, for example, cross-evidence type rules, such as verifying a client's income if it exceeds the maximum allowable threshold. The Household Income evidence type is made of multiple income evidence types such as Dividends, Pension and Wages and Salaries. Though the verification is set up for the income amount of the Household Income evidence

type; the Verification Engine re-evaluates whether the Household Income requires verification when the income of any dependent evidence types, Dividends, Pension and Wages and Salaries, changes.

When you analyze your project requirements it is important to consider for each evidence type if any of the attributes need to be verified and, if so, is it conditional. If it is conditional, you must document the conditional verification rules. Some helpful tips for documenting conditional verifications are:

- The conditional verification rules should be documented in a logical manner.
- Translating the rules into a process flow diagram can make it easier for the developers and testers to understand more complex rules.
- User scenarios can be used to validate the system meets the requirements.
- Document the evidence mapping for the evidence that is checked by the rules.
- Document the information that should be displayed to the user to explain the results of the execution of the conditional rule.

### *Evidence brokering*

You must decide if evidence needs to be shared between cases and persons. Evidence sharing and broadcast settings can be configured. Agencies can define the rules for sharing case evidence per evidence type and per case type.

#### Defining evidence brokering requirements

You must decide how your organization wants evidence to be shared with other organizations.

The Evidence Broker provides a flexible approach to evidence sharing. You should consider the following questions when you define your requirements for the evidence broker:

- Is the same evidence type used on more than one case type? If so, should any changes to this evidence be communicated to other cases? When you document these requirements, capture the case type the evidence is going from and the case type it is going to and whether the evidence types are identical or logically equivalent.
- Will this evidence be shared to a case that auto-accepts or auto-activates evidence, or to the person and prospect person pages, where evidence is never in-edit? If yes, then consider implementing business logic similar to maintenance styles already implemented for person and prospect person evidence. Note: This is important if the implementation needs to support straight-through processing.

To ensure that users of the system can maintain all evidence centrally without having to navigate to different areas of the application, consider that the person or prospect person evidence types be configured on all application cases and integrated cases. If a customer wants to share evidence across cases, the following evidence brokering configurations are recommended:

- Person / Prospect Person to Application Case
- Application Case to Person / Prospect Person
- Integrated Case to Application Case
- Application Case to Integrated Case
- Integrated Case to Person / Prospect Person
- Person / Prospect Person to Integrated Case

#### Adult Social Care evidence brokering configurations

For Adult Social Care, evidence brokering was enabled on the configured evidence types. Identical evidence sharing is configured for the ASC Diagnosis, Incapacity Benefit Product Type, and Practitioner Certificate evidence types. The data is captured on the Adult Social Care application case and shared with the Social Care integrated case.

See the related reference that describes in more detail how evidence brokering is configured in the Adult Social Care system.

### Related reference

[Evidence types configured for Adult Social Care on page 85](#)

Specify the following evidence types to include on the Adult Social Care integrated case. Enable all the evidence types for brokering the evidence.

### Defining the case ownership strategy

You must decide if you will use the default case ownership strategy configured for the Social Care integrated case and Incapacity Benefit product delivery case or if you need to modify it for your project.

#### *Assigning Case Ownership Using Workflow*

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

Note that when a case ownership strategy is defined for a case type, the system transitionally sets the initial owner of a newly created case to the 'Temporary Owner Assignment' work queue while the workflow process determines the initial case owner. Once the workflow process has completed, the case is then reassigned to the case owner determined by the workflow process. There are two system application properties that relate to this work queue.

- The application property, `curam.workflow.logtempownerworkqueueassignment` is used to dictate whether or not case transaction log events that relate to the temporary owner assignment are automatically recorded by the system. The default value is 'NO'. If this value is set to 'YES', a 'User Role Canceled' case transaction log event is automatically recorded when the case is reassigned from the 'Temporary Owner Assignment' work queue.
- The application property, `curam.workflow.displayworkqueueasownerincaseuserrole` is used to dictate whether or not the temporary assignment work queue will appear as a case owner on the User Roles page within a case. The default value is 'NO'.

**Important:** When assigning case ownership to a workflow, administrators must ensure that the case ownership can be resolved through the workflow process that is associated with the case type otherwise the case creation process will fail.

#### *Initial Case Ownership Strategy*

The section describes how the default strategy that determines initial case ownership works. This default strategy can be overridden by administrators using the Ownership Strategy setting described above.

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

1. The administrator of the participant who is the primary client of the integrated case is set as the case owner.
2. If no administrator exists for the primary client, the currently logged in user is set as the case owner.
3. If the participant administrator has no active position within the organization structure, the system assigns case ownership to the currently logged in user.
4. If the currently logged in user has no active position within the organization structure, the user will receive a validation message stating that the case can not be created because no case owner can be identified.

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

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

### **Defining Evidence Management wizards for recording a change in circumstance**

As part of designing a solution, it can be necessary for a change in circumstance to chain together updates to a number of evidence types. An Evidence Management wizard can be used to help the user through the change process. Guidelines on when to introduce an Evidence Management wizard are discussed here.

A change in circumstance can be recorded by the user by directly updating the evidence or through a guided process known as an Evidence Management wizard. You must decide when an Evidence Management wizard is required. This feature contains a wizard with a sequence of steps for capturing the information. If any follow-up actions are required, such as more evidence to be entered based on the reported change, such evidences are listed on the **Guided Change** page. The points that are listed below offer guidance on when an Evidence Management wizard is suitable for recording a change in circumstance.

1. The change involves more than one evidence type,
2. Evidence updates need to happen in sequence, and
3. Conditional evidences need to be entered based on information captured as part of the change in circumstance

### ***Analyzing the product delivery case requirements***

You must define the requirements for the product delivery case product rules and financial processing of benefits.

#### **Product delivery case**

A product delivery case contains all of the information and functions that are needed to determine benefit eligibility and to deliver these benefits to the organization's clients.

#### **Defining eligibility determination processing**

You must define the requirements for the application rules and eligibility determination configurations.

#### ***Assigning Cúram Express Rules to products***

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

#### **Adult Social Care application rules**

You must document the application rules. You can use this information to determine whether you need to modify or extend the existing Incapacity Benefit eligibility and entitlement rules that are used to determine a client's eligibility to receive a monetary benefit.

The Incapacity Benefit Rule Set contains the program recommendation and product delivery eligibility entitlement rules. The rule set contains the main classes and attributes that are used to define the product delivery structure, Incapacity Benefit Product, and decisions for the product delivery eligibility and entitlement, Incapacity Benefit Product Case.

To determine whether a client is eligible to receive a monetary benefit, the system runs the application rule set composed of the following rules:

- Non-Financial Requirements
- Contribution Rules
- Medical Certificate Rules
- Assistance Unit Determination

The Incapacity Benefit Product Type entity indicates whether the benefit type is Sickness Benefit or Invalidity Benefit. If the client passes the Sickness Benefit and Invalidity Benefit eligibility criteria, the benefit amount is calculated by applying the rate structure for the relevant year. By default, the client receives a Sickness Benefit for six months before automatically moving on to the Invalidity Benefit.

#### **Defining display rule requirements**

You must define the decision categories and rules. The Cúram Express Rules Engine contains features to calculate and display decision details on free-form screens for displaying the detailed breakdown of eligibility/entitlement calculations.

When you design your product, you can choose to output such decision details. Unlike the structure of eligibility/entitlement results and key decision factors, the structure of the output data for decision details is product-specific and so the content and layout of the data that is shown on screens must be defined (by using dynamic UIM screens). The Engine uses these product-specific UIM screens to display the decision details to case worker users.

Typically, the rules for calculation of decision details will "sit on top of" the rules for calculating a case's eligibility and entitlement. Whether you follow the best practice recommendation and

layer your rules this way, then you can make changes to the output of decision details for cases while guaranteeing not to affect any case's underlying eligibility/entitlement results. Data that is calculated for decision details never affects Cúram financials or other processing - the data is used for display purposes only.

When you design the decision rules, use a "backwards" perspective to design your product. Namely, you need to start with the end in mind (how case workers navigate decision details, which details to display, and how they are laid out on the screen).

You must identify and name your display categories, and consider the order in which the categories are displayed to a case worker. Moreover, recall that the first category is displayed by default when the user is presented with a row of tabs, and the first category is used when the user expands a coverage period row when viewing details of a determination. It is recommended that you design your first display category to show overall "summary" details of the case's eligibility and entitlement calculations.

The data that populates a display rules page typically comes from eligibility and entitlement rule sets. However, it is possible for a display rules page to show data that isn't derived from eligibility and entitlement. An example of this is sanction recommendations for household members. If a household member is recommended for a sanction, this is accompanied by a reason for this recommendation. Even though this information might not be used in determining eligibility and entitlement, it can be useful to show this information as part of the display rules for the case.

Eligibility and Entitlement use Cúram Express Rules for this functionality. The CER engine supports this process by acting on the rules that are defined in the CER Rules Editor and the evidence that is captured on a case. The flexibility of CER rules means they can also be used to define display rules to convey information about a case to business users. These rules can be as terse or as verbose as they need to be. For more information, see the *Inside Eligibility and Entitlement Using Cúram Express Rules Guide*.

### Adult Social Care display rules

The PD Incapacity Benefit Summary Category and Incapacity Benefit Decision Comparison are configured for the Incapacity Benefit. Each category entry corresponds to a tab on the decision display page. The category defines a display order for each tab on the page.

#### (deprecated) Defining key decision factor rule requirements

You must decide whether your project uses key decision factor rules. Key decision factors are optional on a project. A key decision factor is a piece of information that can help case workers and clients understand key pieces of data that is used in the eligibility and entitlement results of a case. They can be any kind of data that has a direct impact on a client's eligibility and entitlement. This might be derived data, for example, the total household income, or it might be a piece of evidence recorded against the case, like a new medical condition. It could even be an important event, like the birth of a new child. Presenting the information graphically in this way can help case workers to focus on the most relevant information regarding the client's eligibility on a particular date.

Key decision factors are useful on products where the rules are complex and much evidence is captured. If more data is captured, it means that more data can be updated. In such circumstances, being able to see at a glance all the data that was used in coming up with the current determination, and how it changed over time, is an invaluable tool for a case worker. For

instance, it allows them to answer queries much more quickly for clients who might want to know why their benefit payment was lower this month than last month.

The Cúram Express Rules Engine contains features to calculate and display key decision factors - pieces of data that were important in arriving at eligibility and entitlement results.

When you design your product, you can choose to output such factors. The structure of the output data for key decision factors is imposed by the Engine; and since the structure is fixed, the Engine contains a generic set of screens to display the key decision factors to case workers.

Typically, the rules for calculation of key decision factors will "sit on top of" the rules for calculating a case's eligibility and entitlement. Whether you follow the best practice recommendation and layer your rules this way, then you can make changes to the output of key decision factors for cases while guaranteeing not to affect any case's underlying eligibility and entitlement results. Data that is calculated for key decision factors never affects financial processing or any other processing - the data is used for display purposes only.

When you design the key decision factor rules, use a "backwards" perspective to design your product. Namely, start with the end in mind (how case workers benefit from the additional ability to comprehend a case's details through the medium of key decision factors). First describe the results followed by the engine processing that produces those results before finally describing how the data was calculated.

***Defining eligibility determination requirements***

You must define configuration settings at the product level including the decision summary display strategy; determination comparison strategy; open-ended cases setting; and the reassessment strategy.

The related concept describes these configuration settings in more detail.

**Adult Social Care eligibility determination configurations**

For Adult Social Care, the eligibility determination is configured to allow open ended cases and automatically reassess all cases. The full list of settings is provided.

The following Adult Social Care default eligibility determination settings are available:

<b>Name</b>	<b>Default Setting</b>
Allow Open Ended Cases	Yes
Decision Summary Display Strategy	Total weekly monetary entitlement
Determination Comparison Strategy	Compare all user-facing data
Reassessment Strategy	Automatically reassess all cases

**Defining the financial processes**

You can use the following information to consider how to define the financial processing requirements.

***Adult Social Care financial processing***

The Adult Social Care application generates Sickness Benefit and Invalidity Benefit payments based on the financial schedules that are created when a client is found eligible for the benefit product.

For example, a client can be found eligible for the Sickness Benefit for \$200 weekly over a period of 1 month. Payments are then generated for the benefit based on the financial schedule and paid

to the nominee assigned to receive payment for the benefit. The Adult Social Care application uses Cúram Financials to generate financials. The building blocks of financial processing include financial components, financial instruction line items, financial instructions, and financial instruments. They are used to process payments and liabilities, payments received from clients, and in other account maintenance tasks, including payment cancellations.

#### *Defining payment generation requirements*

The delivery pattern, offsets, and financial calendars impact the processing dates of payment generation. You must consider these three areas when you document the payment generation requirements.

#### Defining delivery pattern requirements

A delivery pattern defines the frequency, cover pattern and method by which payments or bills are issued. You can use the following information to consider how to configure the delivery pattern.

A default delivery pattern is assigned to every product on creation. The cover pattern specifies how the payments are issued, for example, in advance, in arrears, once-off.

An offset typically defines the number of days in advance that a payment needs to be processed in order to reach a case nominee on time. For example, check payments might be processed three days in advance so that the nominee receives the check on time whereas an EFT payment might not need to be processed in advance.

When you analyze your project requirements, it is important to consider how payments can be made. For example, if a benefit must be paid weekly on a Monday by Check, then a delivery pattern of "Weekly by Check in advance on Monday" should be associated to the product and set as the default.

In the Adult Social Care application, the default delivery pattern that is associated to the Incapacity Benefit product is "Weekly by Cash in advance on Monday".

#### Defining financial calendar requirements

The financial calendar is used to store payment exclusion dates. You must define the financial calendar requirements.

Payment exclusion dates represent the days on which the organization cannot make payments by using a particular delivery method. For example, if cash payments for benefit payments are normally made on a Monday and next Monday is a public holiday, it will be marked as an exclusion date. A separate calendar exists for each of the following delivery methods:

- Check
- EFT
- Cash
- Voucher
- Invoice

Prepayment requirement is also captured on a Financial Calendar. Prepayment specifies that financial processing occurs on the nearest processing date before the exclusion date.

## 1.5 Building the Adult Social Care application

Use the following information to understand each of the tasks that are involved in building a basic application. You can follow the tasks on the Adult Social Care application to understand how it was built. You can also complete the tasks on a Cúram Platform installation to build the application yourself.

### Configuring the component order for Adult Social Care

You must add Adult Social Care to the server and client component orders so that Adult Social Care is included in the server and client builds.

#### Before you begin

Install a Cúram Application Development Environment (ADE) for application development. The basic development environment consists of the Cúram Application Development Environment (ADE) and the Java IDE.

#### About this task

You need to create a *CustomEnvironment.bat* file for running Cúram from Microsoft Windows or a *CustomEnvironment.sh* file for UNIX. You must not update the *SetEnvironment.bat/sh* files directly to include the additional components. As the *SetEnvironment.bat/sh* files are supplied with Cúram, the files are overridden if you install a new version of Cúram. `%CURAM_DIR%` is the Cúram installation directory, which by default is `C:\Merative\Curam\Development`.

#### Procedure

1. From the `%CURAM_DIR%` directory, create the *CustomEnvironment.bat* and *CustomEnvironment.sh* files.
2. Edit the *CustomEnvironment.bat* file to add the following details to set the component order for Adult Social Care.

```
set CLIENT_COMPONENT_ORDER=AdultSocialCare,IncapacityBenefit,%CLIENT_COMPONENT_ORDER%
set SERVER_COMPONENT_ORDER=
IncapacityBenefit,AdultSocialCare,ASCIntakeConfiguration,%SERVER_COMPONENT_ORDER%
```

3. Edit the *CustomEnvironment.sh* to add the following details to set the component order for Adult Social Care.

```
export CLIENT_COMPONENT_ORDER=AdultSocialCare,IncapacityBenefit,
$CLIENT_COMPONENT_ORDER
export SERVER_COMPONENT_ORDER=
IncapacityBenefit,AdultSocialCare,ASCIntakeConfiguration,$SERVER_COMPONENT_ORDER
```

4. From the `%CURAM_DIR%` directory, run *SetEnvironment.bat*

# Create a user and associated user application view

Create a user and define the user application view. A user application view presents a user with a pre-defined view of the application specific to their user role.

## Configuring a user application view

Complete the following administration tasks to configure a user application view.

### Creating an application home page

Home pages are pages that are tailored to specific roles within an organization, such as case worker, intake worker, or supervisor. Every application has a home page. The home page can be configured to contain pods, announcements, and quick links. Complete the following administration tasks to configure an application home page.

#### Procedure

1. Create the application home page.  
For example, `<page_name>.uim`.
2. Create a properties file for the associated UIM page to include the help text description.  
For example, `<page_name>.properties`.
3. Create a constants file to include the page identifier name.  
For example, `Constants.properties`.
4. From your Cúram web client development installation directory `%CURAM_DIR%\webclient`, run `build client`.  
`%CURAM_DIR%` is the Cúram installation directory, which by default is `C:\Merative\Curam\Development`.
5. Refresh the workspace within the Cúram Application Development Environment (ADE).

### Adult Social Care user home page files

The following are the configuration files that are used for configuring the eligibility worker home page for Adult Social Care.

File name	Location
<code>ASCEligibilityWorkerHome.uim</code>	<code>%CURAM_DIR%\webclient\components\AdultSocialCare\WorkerHomePages</code>
<code>ASCEligibilityWorkerHome.properties</code>	<code>%CURAM_DIR%\webclient\components\AdultSocialCare\WorkerHomePages</code>
<code>Constants.properties</code>	<code>%CURAM_DIR%\webclient\components\AdultSocialCare\WorkerHomePages</code>

**Note:** `%CURAM_DIR%` is the Cúram installation directory which by default is `C:\Merative\Curam\Development`

### Configuring personalized pod pages

You must configure a personalized pod page for each application home page. Personal pages contain multiple pods. The user can alter the content and layout of the personal page by rearranging the pods on the page, by adding more pods or by hiding pods. Complete the following administration tasks to configure personalized pod pages.

#### Procedure

1. Log in to Cúram as a user with administrator credentials.

2. Select the **Administration Workspace** tab, expand the **Shortcuts** menu, and click **User Interface > Personalized Pods Pages**.
3. Click **New Personalized Page**
4. From the **Configure a Personal Page** wizard, complete the following steps.
  - a) From the **Page ID** window, enter the *Page ID* of the UIM to be configured and click **Next**.
  - b) From the **User Role** window, select a role to associate with the page and click **Next**.
  - c) From the **Available Pods** window, select the **PODs** to be displayed on the page and click **Next**.
  - d) From the **Defaults Pods** window, select the **PODs** to be displayed by default and click **Next**.
  - e) From the **Page Layout** window, enter the number of columns to be displayed on the page and click **Next**.
  - f) Click **Save**.

*Pod values for Adult Social Care*

Specify the following values for creating the personalized pods for the Adult Social Care eligibility worker.

Name	Value
<b>Page ID</b>	ASCEligibilityWorkerHome
<b>User Role</b>	ELIGIBILITYROLE
<b>Available Pods</b>	Select the following check boxes: <ul style="list-style-type: none"> <li>• Applications Awaiting Determination</li> <li>• Assigned Applications</li> <li>• Find Client</li> <li>• My Appointments</li> <li>• My Items of Interest</li> <li>• My Queries</li> <li>• Quicks Links</li> <li>• Recent Notifications</li> </ul>
<b>Default Pods</b>	Select All pods.
<b>Page Layout</b>	3

**Configuring a tab**

Content in a section is displayed in a tab, and each section can open multiple tabs, where each tab represents a business object or logical grouping of information. A tab can also be described as a logical grouping of pages. Complete the following administration tasks to configure a tab.

**Procedure**

1. From the **Administration Workspace > User Interface > Tabs**, click **New Tab...**
2. From the **New Tab** window, complete the following fields and click **Save**.

Name	Description
ID	Specify the page name. The tab is displayed when the page is loaded.
Name	The tab name.

Name	Description
Title	The tab title.

### Tab values for Adult Social Care

Specify the following values for creating a tab for the Adult Social Care eligibility worker home page.

Name	Value
ID	ASCEligibilityWorkerHome
Name	Home
Title	Home

### Configuring a section

An application contains a number of sections, which allow quick and easy access to some of the more common tasks and activities that are completed by a user. A user application view can define between one and five sections. For example, Home, Inbox, Case, Outcomes, and Calendar. Each section supports displaying multiple object tabs. For creating a section, you need to create a home and workspace section. The home and workspace sections are mandatory, and a maximum of five sections is advised.

### Procedure

From the **Administration Workspace > User Interface > Sections**, click **New Section...**

#### Configuring a home section

Complete the following administration tasks to configure a home section.

### Procedure

1. From the **New Section** window, specify the following fields and click **Save**.

Name	Description
<b>Section ID</b>	Specify the home section identifier. For example, <i>&lt;application-code&gt;HomeSection</i> .
<b>Section Title</b>	Specify the section title.
<b>Default Page ID</b>	Specify the page identifier. The default page identifier element refers to the page that is opened when the section is selected. The page must be referenced by a tab so that the associated tab is always found for the section.
<b>Hide Tab Container</b>	Select the check box. The hide tab container element indicates that there is only one tab in the section, and the tab bar is not displayed.

2. Select the section.
3. From the **Tabs** tab, click **Add**.
4. From the **Add Tabs** window, enter the tab name and click **Search**.
5. Select the check box for the tab you want to add and click **Save**.

## Home section values for Adult Social Care

Specify the following values for creating a section for the Adult Social Care eligibility worker home section.

Table 12: New Home section values

Name	Value
<b>Section ID</b>	ASCEWAPPHomeSection
<b>Section Title</b>	Home
<b>Default Page ID</b>	ASCEligibilityWorkerHome
<b>Hide Tab Container Indicator</b>	True

Table 13: New tab to search for and select

Name	Value
<b>Tab Name</b>	ASCEligibilityWorkerHome

## Configuring a workspace section

Complete the following administration tasks to configure a workspace section.

## Procedure

1. From the **New Section** window, specify the following fields and click **Save**.

Name	Description
<b>Section ID</b>	Specify the home section identifier, for example <i>&lt;application-code&gt;WorkspaceSection</i> .
<b>Section Title</b>	Specify the section title.
<b>Default Page ID</b>	Specify the page identifier. The default page identifier element refers to the page that is opened when the section is selected. The page must be referenced by a tab so that the associated tab is always found for the section.
<b>Hide Tab Container</b>	Leave the check box as blank. The hide tab container element indicates that there is only one tab in the section, and the tab bar is not displayed.

2. Select the section.
3. From the **Shortcuts Panel** tab, click **New Shortcuts Group**.
4. From the **New Shortcuts Group** window, specify the following fields.

Name	Description
ID	Enter the short group identifier.
Title	Enter the short group title name to be displayed.

5. Select **Edit** to change the shortcuts panel title and click **Save**.

6. Repeat step 3, 4 and 5 to add extra shortcuts groups to the workspace section.
7. From the group that you created, click **New Link**. Specify the following fields and click **Save**.

Name	Description
<b>Link ID</b>	Specify the link identifier. The link identifier must be referenced in the workspace section so that the associated tab is always found. For example, <code>&lt;sc:tab id="&lt;link-id&gt;" /&gt;</code>
<b>Page ID</b>	Specify the page identifier. The default page identifier element refers to the page that is opened when the shortcuts link is selected. The page must be referenced by a tab so that the associated tab is always found for the link.
<b>Title</b>	Specify the title to be displayed for the link.
<b>Open As Modal</b>	Select the check box to open the link as a modal window.

8. Repeat step 7 to add links for each group created.
9. From the **Tabs** tab, click **Add**. Specify the **Tab Name** and click **Search**. Select the tab check box and click **Save**.
10. Repeat step 9 to add the associated tabs for the links added. The page added during a new link creation must be referenced by a tab so that the associated tab is always found for the link.

Workspace section values for Adult Social Care

Specify the following values for creating a section for the Adult Social Care eligibility worker workspace section.

Table 14: New Section values

Name	Value
<b>Section ID</b>	ASCEWAPPWorkspaceSection
<b>Section Title</b>	Clients and Outcomes
<b>Default Page ID</b>	Leave blank.
<b>Hide Tab Container Indicator</b>	Leave the check box blank.

Table 15: New Searches shortcuts group values

Name	Value
ID	Searches
Title	Searches

Table 16: New Searches link values

Link ID	Page ID	Title
PersonSearch	Person_search1	Person...
EmployerSearch	Employer_search1	Employer...

Link ID	Page ID	Title
AllParticipantSearch	Participant_search	All Participants...
sep4		node.sep4.title
Case	Case_search1	Case...
Appeal	Appeal_search	Appeal
sep5		node.sep5.title
EducationalInstituteSearch	EducationalInstitute_search1	Education Institute...
ExternalPartySearch	ExternalParty_search1	External Party...
ExternalPartyOfficeSearch	ExternalPartyOffice_search1	External Party Office...
InformationProviderSearch	InformationProvider_search1	Information Provider...
UtilitySearch	Utility_search1	Utility...

Table 17: New Registration shortcuts group values

Name	Value
ID	ParticipantRegistration
Title	Registration

Table 18: New Registration link values

Link ID	Page ID	Title	Open as Modal
PersonRegistration	RegisterPerson_resolveStartWizard	Person	Yes
ProspectPersonRegistration	RegisterProspectPerson_resolveStartWizard	Prospect Person	Yes
sep6		node.sep6.title	No
EmployerRegistration	RegisterEmployer_resolveStartWizard	Employer	Yes
ProspectEmployerRegistration	RegisterProspectEmployer_resolveStartWizard	Prospect Employer	Yes
sep7		node.sep7.title	No
EducationInstituteRegistration	RegisterEducationalInstitute_resolveStartWizard	Education Institute	Yes
ExternalPartyRegistration	RegisterExternalParty_resolveStartWizard	External Party	Yes
InformationProviderRegistration	RegisterInformationProvider_resolveStartWizard	Information Provider	Yes
UtilityRegistration	RegisterUtility_resolveStartWizard	Utility	Yes

Table 19: New Cases shortcuts group values

Name	Value
ID	Cases
Title	Cases

Table 20: New Cases link values

Link ID	Page ID	Title	Open as Modal
NewCaseQuery	Case_createQuery	New Case Query	Yes
RegisterPersonCreateCase	Case_resolveRegisterPersonAndCreateCase	Register Person and Create Case	Yes

Link ID	Page ID	Title	Open as Modal
AddProspectCreateCase	Case_resolveRegisterProspectPersonAndCreateCase	Add Prospect Person and Create Case	Yes
sep2		node.sep2.title	No
MyCaseQueries	Case_queryList1	My Case Queries	No
MyItemsOfInterest	Organization_listUserBookmark	My Items of Interest	No
sep3		node.sep3.title	No
MyCases	Organization_userCases	My Cases	No
MyRecentlyApprovedCases	Organization_listRecentApprovedCases	My Recently Approved Cases	No
MyRecentlyAssignedCases	Organization_listRecentAssignedCases	My Recently Assigned to Me	No
MyRecentlyViewedCases	Organization_listRecentViewedCases	My Recently Viewed Cases	No

Table 21: Search, case and dynamic evidence tabs

Search Tabs	Cases Tabs	Dynamic Evidence Tabs
PersonSearch	MyCaseQueries	DynamicEvidenceObjectDET240012014010
EmployerSearch	MyItemsOfInterest	DynamicEvidenceObjectDET240022014010
ParticipantSearch	MyCases	DynamicEvidenceObjectDET240032014010
CaseSearch	MyRecentlyApprovedCases	DynamicEvidenceObjectDET240042014010
AppealSearch	MyRecentlyAssignedCases	DynamicEvidenceObjectDET240052014010
EducationalInstituteSearch	MyRecentlyViewedCases	DynamicEvidenceObjectDET240062014010
ExternalPartySearch	CommonIntakeMyApplications	
ExternalPartyOfficeSearch	CommonIntakeApplicationCase	
InformationProviderSearch	CommonIntakeApplicationSearch	
UtilitySearch		

Table 22: Case management tabs

Case Management Tabs	Case Management Tabs	Case Management Tabs	Case Management Tabs
PersonHome	OverPaymentCase	UserSearch	AssessmentHometab
DefaultIntegratedCase	PlanItemView	Utility	BankAccount
EvidenceBroker	ProdProvHome	VerificationsHome	Banks
ExternalParty	ProductDelivery	EvidenceType	CaseMemberHome
ExternalUserHome	ProductDeliveryCases	CaseEvidence	Corrections
ExternalUserSearch	ProspectEmployerHome	IncidentSearch	DefaultProductDelivery
FinancialAdministration	ServicePlanHome	ProductProviderSearch	EducationalInstituteHome
Incident	ServicePlans	ServiceSupplierSearch	EmployerHome
InformationProviderHome	ServiceSupplierHome	PaymentCorrectionCase	
IntegratedCase	ServiceUnitPlanItemView	QueryResults	
IntegratedCasesAdministration	StaticEvidence	UnderPaymentCase	

## Configuring an application banner

An application banner is what is displayed at the top of the page when a user logs in to an application. The four configurable aspects of the banner are the application name, application subtitle, welcome message for the logged in user, and the application menu. Complete the following administration tasks to configure an application banner.

### Procedure

1. From the **Administration Workspace > User Interface > Application Management**, click **New Internal Application**.
2. From the **New Application** window, specify the following fields and click **Save**.

Name	Description
ID	The application code identifier. The application code identifier is linked to the Users entity through the <i>CT_APPLICATIONCODE.ctx</i> code table.
Component	Specify the custom component the application is contained within.
Application	Specify the application user full name.
Welcome Message	Specify the welcome message that is displayed to the user when they view the application workspace.  For example, <i>Welcome %user-full-name</i> .

3. Click **Edit**.
4. Click **Edit Application**, specify the following fields and click **Save**.

Name	Description
Preference Link	Select the preference link check box to enable the display of the preferences on the application banner.
Logout Link	Select the log out link check box to enable the display of <b>logout</b> on the application banner.
Preference Title	Specify the preference title.
Logout Title	Specify the log out title.

5. Select **Application Search > Change Search Type**.
6. From the **Change Search Type** window, select **Single Default Search Page** from the drop-down list and click **Save**.
7. Click **Edit** and specify the following fields and click **Save**.

Name	Description
Default Search Page	Specify a default search page.
Initial Text	Specify the text that is displayed inside the search text field.

8. Select the **Sections** tab and click **Add**.
9. From the **Add Sections** window, enter the section title and click **Search**.
10. Select the check boxes for sections you want to add to the application.

#### *Workspace banner values for Adult Social Care*

Specify the following values for creating an application for the Adult Social Care eligibility worker workspace section.

Table 23: New application values

Name	Value
<b>ID</b>	ASCEWAPP
<b>Component</b>	Cúram
<b>Application</b>	Adult Social Care Eligibility Worker
<b>Welcome Message</b>	Welcome %user-full-name

Table 24: New application values

Name	Value
<b>Preference Link</b>	True
<b>Logout Link</b>	True
<b>Preference Title</b>	Preferences
<b>Logout Title</b>	Log out

Table 25: Search values

Name	Value
<b>Default Search Page</b>	Intake_resolveQuickSearch
<b>Initial Text</b>	Enter Reference Number

Table 26: Sections added

Sections
ASCEWAPPHomeSection
ASCEWAPPWorkspaceSection
DefaultAppCalendarSection
DefaultAppInboxSection

### **Configuring an application code**

An application code is what is used to link a home page to an application. Complete the following administration tasks to configure an application code.

#### **Procedure**

1. Log in to Cúram as a user with the system administrator credentials.
2. Select the **System Configurations** tab, expand the **Shortcuts** menu, and click **Application Data > Code Tables**.

3. From the **Search Criteria** cluster, enter the **Application Code** and click **Search**.
4. From the **Application Code**, click **New Item**.
5. From the **New Code Table Item** window, specify the following fields and click **Save**.

Name	Description
Item Name	Specify the item name, which is the name of the page you want to link the application to.
Technical ID (Code)	Specify the application code.

6. Click **Publish** and click **Yes**.

#### *Application data code table values for Adult Social Care*

Specify the following values for linking the eligibility worker home page to the application code for Adult Social Care.

*Table 27: New application values*

Name	Value
Item Name	ASCEligibilityWorkerHome
Technical ID (Code)	ASCEWAPP

### **Configuring a user**

A user is somebody who is employed by the organization to perform tasks on the system, such as creating and managing cases. Complete the following administration tasks to configure a user.

### **Procedure**

1. Log in to Cúram as a user with the administrator credentials.
2. From the **Administration Workspace > Shortcuts > Users**, click **New User**.
3. From the **New User** window, specify the following fields and click **Save**.

Name	Description
<b>Title</b>	Select the title for the user from the drop-down list. The title is a form of Nomenclature identifying the User's status.
<b>First Name</b>	Specify the first or given name of the user.
<b>Last Name</b>	Specify the common name, or the hereditary name, of the user.
<b>Location</b>	Select the search icon. From the <b>Locations</b> window, select the location from the list. The location is a unique reference for the location within the Organization where a Staff Member (User) is located.
<b>Personal Email</b>	Specify the personal email record for the user.
<b>Username</b>	Specify the name of the user. The name is the unique identifier for the user.
<b>Password</b>	Specify the user password, which is used for logging in to Curam.
<b>Application</b>	Select the home page from the drop-down list. The home page is linked to the user application code, which is stored in the <i>CT_APPLICATION_CODE.ctx</i> file.

Name	Description
<b>Role Name</b>	Select the search icon. From the <b>Security Roles</b> window, select the role. The role is assigned by the system to a Security Group record.
<b>Account Enabled</b>	Select the check box to enable the account. The indicator specifies that an account is enabled or not.

### *User values for Adult Social Care*

Specify the following values for creating the eligibility worker user for Adult Social Care.

Table 28: New user values

Name	Value
<b>Title</b>	Mr
<b>First Name</b>	ELIGIBILITY
<b>Last Name</b>	WORKER
<b>Location</b>	Springfield
<b>Personal Email</b>	financialuser@curamssoftware.com
<b>Username</b>	ascewuser
<b>Password</b>	password
<b>Application</b>	ASCEligibilityWorkerHome
<b>Role Name</b>	ELIGIBILITYROLE
<b>Account Enabled</b>	True

### **Assigning a user to an organization unit and a position**

All users in an organization have a position within an overall organization structure. The organization structure itself can consist of one or more organization units. Complete the following steps to assign a user to an organization unit and a position.

#### **Procedure**

1. From the **Administration Workspace > Shortcuts > Organization > My Organization**, and select the **Structures** tab.
2. From the **Organization Structures** window, select an **Organization Structures** name.
3. Select the **Organization Units** tab and select an **Organization Units** name.
4. Select the **Positions** tab.
5. From the list of positions, select the action menu for the position you want to assign the user to and click **Assign User**.
6. From the **Assign User** window, select the search icon and enter the name in the search criteria. Click **Search**.
7. From the **Search Results**, select the user record and click **Save**.

**Assign the eligibility worker position for Adult Social Care**

Specify the following values for assigning the eligibility worker user to a position for Adult Social Care.

Name	Value
<b>Organization Structures</b>	Midway Social Welfare Plan '03
<b>Organization Units</b>	Board of Directors
<b>Position</b>	Chief Executive Officer
<b>Assign User</b>	ASC ELIGIBILITY WORKER

**Configuring quick links**

Quick links are a collection of links to frequently used application pages, which allow users to complete common actions. For example, searching for a person or accessing a page with a list of the cases that are assigned to the user. Complete the following steps to add quick links to an application view.

**Procedure**

1. From the **Administration Workspace > Shortcuts > User Interface > Quick Links by Application View**.
2. From the application, select **Add Existing**.
3. From the **Add Quick Link To Application** window, select from the available check boxes and click **Save**. It is advised for usability purposes that the number of quick links to not exceed 7.

**Quick links values for Adult Social Care**

Specify the following quick links to be displayed on the eligibility worker home page quick link pod.

Name	Value
<b>Application</b>	ASCEligibilityWorkerHome
<b>Quick Links</b>	Select the following quick links: <ul style="list-style-type: none"> <li>• <b>My Applications</b></li> <li>• <b>My Cases</b></li> <li>• <b>Register a Person</b></li> <li>• <b>Register a Prospect Person</b></li> <li>• <b>Search for a Case</b></li> <li>• <b>Search for a Person</b></li> <li>• <b>Change my Password</b></li> </ul>

**Saving your user application view configurations**

Save your changes to the user application view configurations.

**Quick reference for saving user application view files**

Reference table for all files that are required for saving your user application views

The following descriptions detail each column of the reference table and what their purpose is.

- **Task Name:** The name of the current task.

- **Data Contained In:** The relevant file or files in the `%CURAM_DIR%\EJBServer\build\dataextractor` folder, which contains data that you need to extract for the step. `%CURAM_DIR%` is the Cúram installation directory, which by default is `C:\Merative\Curam\Development`.
- **Files To Be Copied From Data Extractor:** A list of files that the user must copy from the `%CURAM_DIR%\EJBServer\build\dataextractor` folder for each task.
- **Files To Be Created:** A list of files that need to be created by the user for each task.

Table 29: User Application Files Reference Table

Task Name	Data Contained In	Files To Be Copied From Data Extractor	Files To Be Created
Saving the personalized pod page configurations	<code>PAGECONFIG.dmx</code> <code>USERPAGECONFIG.dmx</code>	<code>PAGECONFIG.dmx</code> <code>USERPAGECONFIG.dmx</code>	
Creating the application view configuration files	<code>APPRESOURCE.dmx</code>		<code>ASCEWAPP.app</code> <code>ASCEWAPP.properties</code> <code>ASCEWAPPHomeSection.sec</code> <code>ASCEWAPPHomeSection.properties</code> <code>ASCEWAPPWorkspaceSection.sec</code> <code>ASCEWAPPWorkspaceSection.prope</code> <code>ASCEWAPPWorkspaceSectionShortc</code> <code>ASCEWAPPWorkspaceSectionShortc</code> <code>ASCEligibilityWorkerHome.tab</code> <code>ASCEligibilityWorkerHome.prope</code>
Saving the application code configurations	<code>CODETABLEITEM.dmx</code>		<code>CT_APPLICATION_CODE.ctx</code>
Saving the user configuration files	<code>USERS.dmx</code> <code>USERLINK.dmx</code> <code>POSITIONHOLDERLINK.dmx</code>	<code>USERS.dmx</code> <code>USERLINK.dmx</code> <code>POSITIONHOLDERLINK.dmx</code>	
Saving the user quick links configurations	<code>QUICKLINK.dmx</code> <code>QUICKLINKAPPLICATION.dmx</code>	<code>QUICKLINK.dmx</code> <code>QUICKLINKAPPLICATION.dmx</code>	
Creating an application home page			<code>ASCEligibilityWorkerHome.uimAS</code>

### Updating unique identifiers in configuration files

Updating Unique IDs in configuration files.

#### About this task

A key step in saving your configuration files involves updating the unique identifiers inside the DMX and code table files. Updating the generated identifier attributes to unique numbers ensures that all of the content that you created in the application is saved and remains in the application. Changing the identifiers is necessary as all unsaved data is wiped from the database during a rebuild. Finding the identifier to be changed in your configuration file can be done quickly as the identifier element typically contains the suffix *ID* at the end of its name. It is important to note that there can be multiple identifiers present in a configuration file. Changing each identifier number to one specified in the valid range and ensuring that no duplicate numbers are present ensures that your content is saved. This process occurs over several of the steps but it never changes.

### Saving the personalized pod pages configurations

Complete the following steps to extract the specified personalized pod page DMX files from the database and put them under source control.

#### Procedure

1. Extract the DMX files from the database with the **build extractdata** build target.
2. From the `%CURAM_DIR%\EJBServer\build\dataextractor` directory, copy the following DMX files.

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

Name	Description
<code>PAGECONFIG.dmx</code>	Stores page templates for the personalized pages. The identifiers are referenced from the <code>CT_PodType.ctx</code> code table.
<code>USERPAGECONFIG.dmx</code>	Stores user-specific settings for the personalized pages, including default settings by user role. The identifiers are referenced from the <code>CT_PodType.ctx</code> code table.

3. Inside `PAGECONFIG.dmx` and `USERPAGECONFIG.dmx`, you must find the attribute with `name="config"` and change its `type` from `blob` to `text`.
4. The second step is to replace all `<` symbols with `&lt;` and replace all `>` symbols with `&gt;`;
5. Inside `USERPAGECONFIG.dmx`, you must find the record with a `userPageConfigID` value of `4500` and change its `pageid` from `ASCEligibilityWorkerHome` to `ASCEligibilityWorkerHome2`. This change avoids the Adult Social Care user workspace configuration being overridden by an Intake user workspace configuration.
6. From `%CURAM_DIR%\EJBServer`, run **build database**.
7. Save your configuration files.

#### Personalized pod values for Adult Social Care

The eligibility worker user home page configuration files specific for Adult Social Care.

File name	Description	Location
<code>UserPageConfig.dmx</code>	Stores page templates for the personalized pages.	<code>%CURAM_DIR%\EJBServer\components\AdultSocialCare\data\initial</code>
<code>PageConfig.dmx</code>	Stores user-specific settings for the personalized pages, including default settings by user role.	<div style="border: 1px solid blue; background-color: #e6f2ff; padding: 5px;"> <p><b>Note:</b> <code>%CURAM_DIR%</code> is the Cúram installation directory, which by default is <code>C:\Merative\Curam\Development</code>.</p> </div>
<code>CT_PodType.ctx</code>	This file is included inside the existing <code>%CURAM_DIR%\EJBServer\components\core\codetable</code> folder so it is not necessary to copy it into that directory.	

### Creating the application view configuration files

The section, tab, and application management configurations are stored in *APPRESOURCE.dmx*. The associated *blob* files that are extracted from the database need to be converted into the associated *.app*, *.sec*, *.ssp*, *.tab*, *.nav*, *.mnu*, and *.properties* files.

#### Procedure

1. Extract the DMX files from the database with the *extractdata* build target.
2. From the `%CURAM_DIR%\EJBServer\build\dataextractor` directory, search for the `<application-code>` in the *APPRESOURCE.dmx* file.  
`%CURAM_DIR%` is the Cúram installation directory, which by default is `C:\Merative\Curam\Development`.
3. For each of the rows that are returned in the search results, the *blob* files are specified in the content attribute. Extract all the application resource *blob* files referenced.
4. Rename the *blob* files and store in the `%CURAM_DIR%\EJBServer\components\COMPONENT_NAME\clientapps` directory. XML configuration files that are included with Curam exist in the `%CURAM_DIR%\EJBServer\components\COMPONENT_NAME\tab` directory. The row names that are specified in the *APPRESOURCE.dmx* file are specified here. The names that the associated application resource files need to be renamed to are specified. The ID in the files needs to be the same name as the file name.

Name	Rename File To
<code>curam.config.app.&lt;application-code&gt;.app</code>	<code>&lt;application-code&gt;.app</code>
<code>&lt;application-code&gt;.properties</code>	<code>&lt;application-code&gt;.properties</code>
<code>curam.config.sec.&lt;application-code&gt;HomeSection.sec</code>	<code>&lt;application-code&gt;HomeSection.sec</code>
<code>&lt;application-code&gt;HomeSection.properties</code>	<code>&lt;application-code&gt;HomeSection.properties</code>
<code>curam.config.sec.&lt;application-code&gt;WorkspaceSection.sec</code>	<code>&lt;application-code&gt;WorkspaceSection.sec</code>
<code>&lt;application-code&gt;WorkspaceSection.properties</code>	<code>&lt;application-code&gt;WorkspaceSection.properties</code>
<code>curam.config.ssp.&lt;application-code&gt;WorkspaceSection_SSP</code>	<code>&lt;application-code&gt;WorkspaceSectionShortcuts.ssp</code>
<code>&lt;application-code&gt;WorkspaceSection_SSP.properties</code>	<code>&lt;application-code&gt;WorkspaceSectionShortcuts.properties</code>

5. Search for the `<page-name>` in the *APPRESOURCE.dmx* file.
6. Repeat step 3.

Name	Rename File To
<code>curam.config.tab.&lt;page-name&gt;</code>	<code>&lt;page-name&gt;.tab</code>
<code>&lt;page-name&gt;.properties</code>	<code>&lt;page-name&gt;.properties</code>
<code>curam.config.mnu.&lt;page-name&gt;Menu</code>	<code>&lt;page-name&gt;Menu.mnu</code>

Name	Rename File To
<page-name>Menu.properties	<page-name>.properties

- From %CURAM\_DIR%\EJBServer, run **build inserttabconfiguration**.
- Save your configuration files.

### User application view files for Adult Social Care

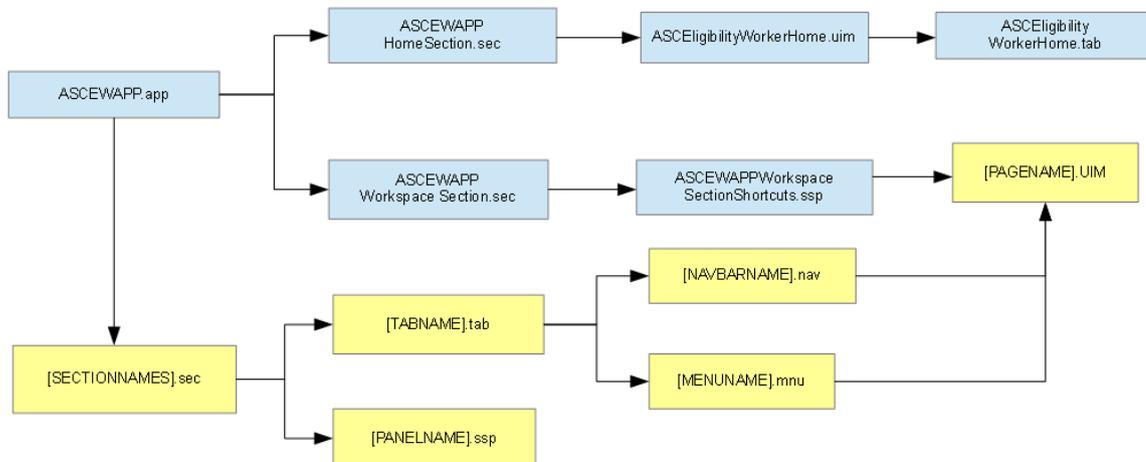
The following specify the configuration files that configure the Adult Social Care eligibility worker application view.

File name	Description	Location
ASCEWAPP.app and the associated properties file.	<p>For each section referenced in the ASCEWAPP.app file specific to Adult Social Care, the related section .sec file must be defined to include the tabs available in each section.</p> <ul style="list-style-type: none"> <li>The following workspace sections are defined.                             <ul style="list-style-type: none"> <li>Home</li> <li>Clients and Outcomes</li> <li>Inbox</li> <li>Calendar</li> </ul> </li> </ul> <p>The <b>Home</b> and <b>Clients and Outcomes</b> sections are customized for Adult Social Care. The remaining sections, <b>Inbox</b>, and <b>Calendar</b>, use the default sections as defined in the core application. A maximum of 5 sections is advised.</p>	<p>%CURAM_DIR%\EJBServer                      \components\AdultSocialCare\tab                      \ApplicationViews</p> <div style="border: 1px solid blue; padding: 5px; background-color: #e6f2ff;"> <p><b>Note:</b> %CURAM_DIR% is the Cúram installation directory, which by default is C:\Merative\Curam\Development.</p> </div>
ASCEWAPPHomeSection.sec and the associated properties file.	<p>The Adult Social Care <b>Home</b> section references the eligibility home page.</p>	
ASCEWAPPWorkspaceSection.sec and the associated properties file.	<p>The <b>Clients and Outcomes</b> section specifies the associated tabs, including the core tabs, the common intake tabs, the evidence tabs, and the Adult Social Care tabs.</p> <p>The Adult Social Care shortcuts panel is also specified.</p> <p>The persons information is specified in the <b>Clients and Outcomes</b> section.</p>	
ASCEWAPPWorkspaceSection.sec and the associated properties file.	<p>The shortcut links in the <b>Clients and Outcomes</b> section are defined. The .tab file (id) and .uim (page-id) must also be defined when the shortcut is specified.</p>	

File name	Description	Location
<p><i>ASCEligibilityWorkerHomeTab</i> and the associated properties file.</p>	<p>Defines the eligibility worker home tab. The Adult Social Care eligibility worker home page is specified. The page must be referenced in a tab so that the associated tab is always found for the section.</p> <p>The context panel and tab navigation is not configured as part of Adult Social Care home page.</p> <p>The context panel can be configured to display a specific UIM page that displays common information for the tab that is always viewable.</p> <ul style="list-style-type: none"> <li>• <code>&lt;contextpanel&gt;.uim</code></li> </ul> <p>The tab navigation can be configured to group UIM pages as part of a tab can be navigated to within a tab.</p> <ul style="list-style-type: none"> <li>• <code>&lt;navigation&gt;.nav</code></li> </ul>	<p><code>%CURAM_DIR%\EJBServer\components\AdultSocialCare\tab\EligibilityWorkerTabs</code></p>
<p><i>ASCEligibilityWorkerHomeTab</i> and the associated properties</p>	<p>Defines the content of a tab actions menu.</p>	

*Adult Social Care user application files link diagram*

The following image displays the associated links between the user application view files that are created for the Adult Social Care eligibility worker user.



**Saving the application code configurations**

Extract the content from the *CODETABLEITEM.dmx* file and add it to the *CT\_APPLICATION\_CODE.ctx* code table to save the application code configurations.

**Procedure**

1. Extract the DMX files from the database with the *extractdata* build target.
2. From the *%CURAM\_DIR%\EJBServer\build\dataextractor\CODETABLEITEM.dmx* file, search for the application code configured.

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

3. If there is no pre-existing `CT_APPLICATION_CODE.ctx` code table in the `%CURAM_DIR%\EJBServer\components\COMPONENT_NAME\codetable` directory, create a new code table and add the content from the `CODETABLEITEM.dmx` file. If the code table exists, add the content to the existing code table.
4. From `%CURAM_DIR%\EJBServer`, run **build ctgen database**.
5. From `%CURAM_DIR%\webclient`, run **build client**.
6. Save your configuration files.

#### User application code file for Adult Social Care

The following is the configuration file used for configuring the application code for Adult Social Care.

File name	Location
<code>CT_APPLICATION_CODE.ctx</code>	<code>%CURAM_DIR%\EJBServer\components\AdultSocialCare\codetable</code>

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

#### Saving the user configuration files

Extract the specified DMX files from the database and put them under source control.

#### Procedure

1. Extract the DMX files from the database with the `extractdata` build target.
2. From the `%CURAM_DIR%\EJBServer\build\dataextractor` directory, copy the following DMX files.

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

File name	Description
<code>USERS.dmx</code>	Stores user information.
<code>USERLINK.dmx</code>	Stores a link between the user name and user ID.
<code>POSITIONHOLDERLINK.dmx</code>	Stores a link between a position and a user.

3. From `%CURAM_DIR%\EJBServer`, run **build database**.
4. Save your configuration files.

### User configuration files for Adult Social Care

The following are the configuration files that are used for configuring the eligibility worker for Adult Social Care.

**Note:** The eligibility worker is the first point of contact for the organization to determine its clients' needs. Then, provide the final determination and authorization of benefits to a client. Extra supervisor rules can be set up for the project, however, is not something that was carried out for Adult Social Care.

File name	Location
<i>USERS.dmx</i>	%CURAM_DIR%\EJBServer\components\AdultSocialCare\data
<i>USERLINK_EO.dmx</i>	\demo
<i>POSITIONHOLDERLINK.dmx</i>	<b>Note:</b> %CURAM_DIR% is the Cúram installation directory, which by default is C:\Merative\Curam\Development.

### Saving the user quick links configurations

Extract the specified DMX files from the database and put them under source control.

#### Procedure

1. Extract the DMX files from the database with the *extractdata* build target.
2. From the %CURAM\_DIR%\EJBServer\build\dataextractor directory, copy the following DMX files.

**Note:** %CURAM\_DIR% is the Cúram installation directory, which by default is C:\Merative\Curam\Development.

File name	Description
<i>QUICKLINK.dmx</i>	When a user logs in to the system, any quick links that are associated with their role are displayed on their home page. Quick links are shortcut links, which can be included within a user role home page <b>Quick Links</b> pod. All quick links are defined in the <i>QUICKLINK.dmx</i> file.
<i>QUICKLINKAPPLI</i>	Defines the quick links for the application. The quick links act as shortcuts for users for commonly accessed parts of an application, for example, the Financial or Case Worker Application.  A <b>Quick Links</b> pod is displayed on a user role home page. The quick link must be associated with the user role for it to display in the <b>Quick Links</b> pod. For each quick link to be displayed in the <b>Quick Links</b> pod of a user role home page, an entry must be added to <i>QUICKLINKAPPLICATIONLINK.dmx</i>

3. From %CURAM\_DIR%\EJBServer, run **build database**.
4. Save your configuration files.

### User quick links files for Adult Social Care

The following are the DMX files specific for creating quick links for the Adult Social Care eligibility worker user home page.

File name	Description	Location
<i>QUICKLINK.dmx</i>	All quick links are defined in the <i>QUICKLINK.dmx</i> file.	%CURAM_DIR%\EJBServer\components\Intake\data\initial
	This file is already included inside the existing %CURAM_DIR%\EJBServer\components\core\data\demo folder so it is not necessary to copy it into that directory.	%CURAM_DIR%\EJBServer\components\core\data\demo
<i>QUICKLINKAPPLICATION.dmx</i>	Defines the quick links for the application.	%CURAM_DIR%\EJBServer\components\AdultSocialCare\data\initial

**Note:** %CURAM\_DIR% is the Cúram installation directory, which by default is C:\Merative\Curam\Development.

## Configuring evidence

Complete the following tasks to configure and view the evidence entities that are used to hold Adult Social Care case data.

### Configure the evidence entities for Adult Social Care

A Dynamic Evidence Type is a logical grouping of related attributes about which an organization wants to record information in respect of a Case. These evidence types are configured and maintained by the Cúram Administration Suite.

#### Procedure

1. Select the **Administration Workspace** tab, expand the **Shortcuts** menu, and select **Rules and Evidence > Dynamic Evidence**
2. View the list of Adult Social Care evidence types

Entity	Description
ASC Household Member	This entity contains details of an individual who lives in the household. Personal information is captured for household members through the system registration process. Additional information such as citizen status, veteran status and ethnic details, may be captured on the household record. etc
ASC Living Arrangement	This entity describes the living arrangement of the client, for example, if they live at home, are homeless, or live in an institution.

Entity	Description
ASC Alien	If the client is an alien, this entity holds information in relation to the alien. Alien types include refugee, asylee, and parolee. The alien must be determined an eligible alien to receive assistance from certain programs.
Practitioner Certificate	The Practitioner Certificate entity is a representation of a certificate that is received by a medical professional. It contains the from and to dates that the certificate covers, the date of examination, the medical practitioner's name, and it is also signed by the practitioner. In evidence terms, the Practitioner Certificate is a parent of the ASC Diagnosis entity.
ASC Diagnosis	The ASC Diagnosis entity contains the diseases or ailments that are suffered by the client. It also contains the date of examination. In evidence terms, the ASC Diagnosis entity is a child of Practitioner Certificate. It's possible for the client to suffer from one or more ailments, in which case one diagnosis record is created for each ailment.
Incapacity Benefit Product Type	The Incapacity Benefit Product Type entity determines whether the client receives Sickness Benefit or Invalidity Benefit. It is defaulted to Sickness Benefit for the first 182 days and automatically moves to Invalidity Benefit if the client is still receiving benefit.

### What to do next

From the Evidence Type list window evidence entities are created and maintained. Full details of all actions and processes that are associated with evidence is documented in the Configuring Dynamic Evidence guide.

### ***Saving your evidence configuration files.***

The files are specified for saving the evidence configurations.

#### **Quick reference for saving evidence configuration files**

Reference table for all files that are associated with saving your evidence configuration files.

The following descriptions detail each column of the reference table and what their purpose is.

- Task Name: The name of the current task.
- Data Contained In: The relevant file or files in the `%CURAM_DIR%\EJBServer\build\dataextractor` folder, which contains data that you need to extract for the step. `%CURAM_DIR%` is the Cúram installation directory, which by default is `C:\Merative\Curam\Development`.
- Files To Be Copied From Data Extractor: A list of files that the user must copy from the `%CURAM_DIR%\EJBServer\build\dataextractor` folder for each task.
- Files To Be Created: A list of files that need to be created by the user for each task.

The following descriptions detail each column of the reference table and what their purpose is.

- Task Name: The name of the current task.
- Data Contained In: The relevant file or files in the `%CURAM_DIR%\EJBServer\build\dataextractor` folder, which contains data that you need to extract for the step.

- Files To Be Copied From Data Extractor: A list of files that the user must copy from the %CURAM\_DIR%\EJBServer\build\dataextractor folder for each task.
- Files To Be Created: A list of files that need to be created by the user for each task.

Table 30: Evidence Configuration File Reference Table

Task Name	Files To Be Copied From Data Extractor	Data Contained In
Saving your evidence files	CT_EvidenceType.ctx	%CURAM_DIR%
	CT_TemporalEvidenceTypeApproval.ctx	\EJBServer\components \AdultSocialCare \codetable
	EVIDENCETYPEDEF.dmx	%CURAM_DIR%
	EVIDENCETYPEVERSIONDEF.dmx	\EJBServer\components \AdultSocialCare\data
	EVIDENCETYPEDEFINITION.dmx	\initial
	EVIDENCERULESETDEF.dmx	
	SECURITYGROUP.dmx	
	SECURITYGROUPSID.dmx	
	SECURITYIDENTIFIER.dmx	
	RULEOBJECTPROPAGATORCONFIG.dmx	
	CREOLERULESET.dmx	
	LOCALIZABLETEXT.dmx	
	TEXTTRANSLATION.dmx	
	CREOLERULESETCATEGORYLINK.dmx	%CURAM_DIR%
		\EJBServer\components \IncapacityBenefit\data \initial
<b>Clob files</b>	%CURAM_DIR%	
Metadata_<EvidenceType>_<EffectiveDate>.xml	\EJBServer\components \COMPONENT_NAME\data \initial\clob	
<b>Blob files</b>	%CURAM_DIR%	
<EvidenceType>RuleSet.xml	\EJBServer\components \COMPONENT_NAME\data \initial\blob	
<EvidenceType>DataRuleSet.xml		
.tab and .nav		
<b>All relevant .tab and .nav files</b>	%CURAM_DIR%\EJBServer \build\data\DEExtract \COMPONENT_NAME\tab	

### Extracting configured evidence artifacts

When evidence is configured by using the editor and that version is activated, all artifacts that are associated with the configured type need to be extracted from the database.

#### Procedure

1. From `%CURAM_DIR%\EJBServer\components\DynamicEvidence\script` directory, run the **extractDynamicEvidenceConfiguration** batch utility and specify the necessary parameters. See the following table for the list of mandatory parameters:

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

Parameter	Description
-Dcomponent	The component to which the configured dynamic evidence belongs. For example, COMPONENT_NAME
-DlowerKey	The lower or starting key for the keyrange that is used to generate replacement primary keys for extracted database records.
-DupperKey	The upper or end key for the keyrange that is used to generate replacement primary keys for extracted database records.
-DetExtractionList	A semi-colon delimited list of dynamic evidence types to be extracted. For example, "testET1;testET2".

2. When the extraction is complete, all artifacts are available in the `%CURAM_DIR%\EJBServer\build\data\DEExtract\COMPONENT_NAME` directory.

### Saving your evidence files

All the artifacts that are associated with extracted evidence need to be put under source control.

#### Procedure

1. From the `%CURAM_DIR%\EJBServer\build\data\DEExtract\COMPONENT_NAME` directory, copy the following files.

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

Table 31: Code tables

File name	Description	Location
<i>CT_EvidenceType.ctx</i>	<p>The <i>CT_EvidenceType.ctx</i> code-table contains the codes for the newly added dynamic evidence types. The code values start at "DET0000256", but it's advised to update them in line with the keyrange used.</p> <p>If there is no pre-existing code table file in <code>%CURAM_DIR%\EJBServer\components\COMPONENT_DIR\codetable</code>, copy the modified code table files to this location. If they do exist, add the modified codes to the respective code-tables.</p>	<p><code>%CURAM_DIR%\EJBServer\build\data\DEExtract\COMPONENT_NAME\codetable</code></p>
<i>CT_TemporalEvidenceTypeApproval.ctx</i>	<p>The <i>CT_TemporalEvidenceTypeApproval.ctx</i> is similar to <i>CT_EvidenceType.ctx</i>. It's used specifically for setting up evidence approvals in the Administration Application. Again, update the codes in line with the keyrange.</p>	

Table 32: DMX files

File name	Description	Location
<p><i>EVIDENCERULESETDEF.DMX</i>  <i>EVIDENCETYPEDEF.dmx</i>  <i>EVIDENCETYPEDEFINITION.dmx</i>  <i>EVIDENCETYPEVERSIONDEF.DMX</i></p>	<p>The evidence types and evidence type version definitions are governed by the entries in these DMX files:</p> <p>For an activated dynamic evidence version, <i>EVIDENCERULESETDEF.dmx</i> maps the evidence type to its associated propagator. For in-edit dynamic evidence, <i>EVIDENCERULESETDEF.dmx</i> also points to in-edit data and processing rule sets.</p> <p>Both <i>EVIDENCETYPEDEF.dmx</i> and <i>EVIDENCETYPEDEFINITION.dmx</i> reference the evidence type codes as specified in <i>CT_EvidenceType.ctx</i>. If these values were modified in line with the keyrange, the references in both these files also need to be updated.</p> <p><i>EVIDENCETYPEVERSIONDEF.dmx</i> points at the evidence metadata files in <code>%CURAM_DIR%\EJBServer\component\COMPONENT_NAME\data\initial\clob</code>.</p>	<p><code>%CURAM_DIR%\EJBServer\build\data\DEExtract\COMPONENT_NAME\data\initial</code></p>

File name	Description	Location
<p><i>SECURITYGROUP.dmx</i> <i>SECURITYGROUPSID.dmx</i></p>	<p>These DMX files store the security for the extracted evidence types.</p> <p>The create, modify, and read security identifiers (SIDs) for each dynamic evidence type can be found in <i>SECURITYIDENTIFIER.dmx</i>.</p> <p>A security group is generated for each evidence type configured. These new groups can be found in <i>SECURITYGROUP.dmx</i>.</p> <p>The SIDs are linked to their respective security group in <i>SECURITYGROUPSID.dmx</i>. They are also linked to the generic security group, <i>EVIDENCEGROUP</i>.</p>	
<p><i>RULEOBJECTPROPAGATORCONFIG.dmx</i></p>	<p>The propagator for evidence types are configured in <i>RULEOBJECTPROPAGATORCONFIG.dmx</i>. This DMX file points at the XML propagator configuration files in the <code>%CURAM_DIR%\EJBServer\components\COMPONENT_NAME\data\initial\clob</code> directory.</p>	
<p><i>CREOLERULESET.dmx</i> <i>CREOLERULESETCATEGORYLINK.dmx</i></p>	<p>The rule sets associated with extracted evidence types are configured in <i>CREOLERULESET.dmx</i>. This DMX file points at the rule set XML files in <code>data\initial\blob</code>.</p> <p>At a minimum, there are 2 rule sets. These rule sets are <code>&lt;EvidenceType&gt;RuleSet.xml</code> and <code>&lt;EvidenceType&gt;DataRuleSet.xml</code>. More rule sets are present if, for example, a validation rule set was specified during configuration or if a calculated attribute was added to the dynamic evidence type.</p> <p>When the rule sets are being put under source control, move them to the <code>%CURAM_DIR%\EJBServer\COMPONENT_NAME\CREOLE_Rule_Sets</code> directory and update the references in <i>CREOLERULESET.dmx</i> to point at their new location.</p> <p>The <i>CREOLERULESETCATEGORYLINK.dmx</i> categorizes each of the generated rule sets.</p>	
<p><i>LOCALIZABLETEXT.dmx</i> <i>TEXTTRANSLATION.dmx</i></p>	<p>The <i>LOCALIZABLETEXT.dmx</i> and <i>TEXTTRANSLATION.dmx</i> files contain descriptions of the rule set propagators.</p>	
<p><i>APPRESOURCE.dmx</i></p>	<p>The <i>APPRESOURCE.dmx</i> contains entries for all the <code>.tab</code> and <code>.nav</code> files, together with their respective properties. As all <code>.tab</code> and <code>.nav</code> files, and their properties, belong in the <code>%CURAM_DIR%\EJBServer\COMPONENT_NAME\tab</code> directory, the entries in <i>APPRESOURCE.dmx</i> can be deleted.</p>	

Table 33: Clob data files

File name	Description	Location
<code>Metadata_&lt;EvidenceType&gt;_&lt;EffectiveDate&gt;.xml</code>	<p>The <code>data\EffectiveDate</code> directory contains the metadata files for the extracted dynamic evidence types.</p> <p>The naming convention of these metadata files is <code>Metadata_&lt;EvidenceType&gt;_&lt;EffectiveDate&gt;.xml</code> where <code>EffectiveDate</code> is the effective date of the evidence type version.</p> <p>An example of a dynamic evidence metadata file is <code>Metadata_ACMEEvidence_20140101.xml</code></p>	<pre>%CURAM_DIR %\EJBServer \components \COMPONENT_NAME \data\initial \clob</pre>

Table 34: Blob data files

File name	Description	Location
<pre>&lt;EvidenceType&gt;RuleSet &lt;EvidenceType&gt;DataRuleSet.xml .tab .nav</pre>	<p>The <code>blob</code> directory contains:</p> <ol style="list-style-type: none"> <li>1. Rule sets</li> <li>2. <code>.tab</code> and <code>.nav</code> files for each dynamic evidence type and their associated properties</li> <li>3. Two properties files for each dynamic evidence type</li> </ol>	<pre>%CURAM_DIR %\EJBServer \components \data\initial /blob</pre>

Table 35: Tab files

File name	Description	Location
<code>.tab</code> and <code>.nav</code> files	<p>The <code>tab</code> directory contains contributions to the existing section files from the newly configured dynamic evidence types.</p> <p>Copy the <code>.tab</code> and <code>.nav</code> files, and their associated properties, to the <code>%CURAM_DIR%\EJBServer\COMPONENT_NAME\tab</code> directory before they are put under source control.</p> <p><b>Section Contributions</b></p> <p>Contributions are made to the following sections by the configured dynamic evidence types</p> <ol style="list-style-type: none"> <li>1. <code>AUDITCOAPPSection.sec</code></li> <li>2. <code>AUDITORAPPSection.sec</code></li> <li>3. <code>DefaultAppSection.sec</code></li> <li>4. <code>FINAPPSection.sec</code></li> <li>5. <code>INVESTRAPPSecion.sec</code></li> <li>6. <code>SUPERAPPSection.sec</code></li> </ol>	<pre>%CURAM_DIR% \EJBServer\build \data\DEExtract \COMPONENT_NAME \tab</pre>

2. From %CURAM\_DIR%\EJBServer, run **build ctgen database**.
3. From %CURAM\_DIR%\webclient, run **build client**.
4. Save the configuration files.

## Configuring products

A product is the mechanism for providing financial benefit to a client. You must configure a product so that integrated cases and product delivery cases can be created.

### Creating code tables

Create code tables to hold the values for the drop-down lists that are needed to configure an application case or evidence type.

#### Procedure

1. Create the following code table as they need to be configured to view the content in the *Evidence Category* drop-down lists from the **Add Evidence Type** window.

File name	Description
CT_EvidenceCategory	This code table defines the evidence categories for each evidence type.

2. Create the following code tables as they need to be configured to view the content in the *Ongoing Integrated Case Type* and *Product Delivery Type* drop-down lists from the **New Program** window.

File name	Description
CT_ApplicationType	This code table defines the type of integrated case that is created to process an application for this program. This code table is only applicable if Curam is selected as the case processing system.
CT_ProgramType.ct	This code table defines the type of program that is created.

### Code tables created for Adult Social Care

The locations of the Adult Social Care code table files are specified.

File name	Location
CT_EvidenceCategory	%CURAM_DIR%\EJBServer\components\AdultSocialCare\codetable
CT_ApplicationType	<b>Note:</b> %CURAM_DIR% is the Cúram installation directory, which by default is C:\Merative\Curam\Development.
CT_ProgramType.ct	

### Configuring a product

On the authorization of an application case, an integrated case and a product delivery case are created based on how they are configured in the product.

#### Related tasks

[Analyzing the integrated case requirements on page 41](#)

You must decide what evidence types are associated with the integrated case. For each evidence type, you must decide how the evidence is updated, how verifications are configured, and how the evidence is brokered between cases and persons. It is also important to define the requirements for the integrated case ownership strategy.

[Analyzing the product delivery case requirements on page 50](#)

You must define the requirements for the product delivery case product rules and financial processing of benefits.

### Configuring integrated cases

Configure an integrated case to hold the evidence that is captured in your intake questionnaire and add the required evidence types. Enable sharing for the specific evidence types for which you want to share evidence between cases.

#### Configuring an integrated case

Complete the following steps to configure an integrated case.

### Procedure

1. Log in to Cúram as a user with the administrator credentials.
2. Select the **Administration Workspace** tab, expand the **Shortcuts** menu, and click **Case > Integrated Cases**.
3. Click **New Integrated Case Type...** From the **New Integrated Case Type** window, specify the following fields and click **Save**.

Name	Description
Integrated Case Type	On creation, the integrated case type code is automatically inserted into the <i>ProductCategory</i> and <i>CaseCatTypeCode</i> code tables.
Determine Translator Required	Specify whether the system automatically determines whether a translator is needed for case participants.
Display in Participant Programs List	Specify whether cases of this integrated case type are displayed in the participants programs list.
Display in Citizen Self Service	Specify whether cases of this integrated case type are displayed in citizen self service.
Display in My Cases filter	Specify whether cases of this integrated case type are displayed in the My Cases filter.
Display in Case Search filter	Specify whether cases of this integrated case type are displayed in the Case Search filter.
Appealable	Specify whether the integrated case can be appealed or not. A false value indicates that the integrated case is appealable.
Members Only for Contact Log	Not applicable.
Ownership Strategy	Not applicable.

### Integrated case values for Adult Social Care

Specify the following integrated case values for Adult Social Care.

Name	Value
Integrated Case Type	Social Care
Determine Translator Required	Leave blank.
Display in Participant Programs List	Select the check box.

Name	Value
Display in Citizen Self Service	Leave blank.
Display in My Cases filter	Select the check box.
Display in Case Search filter	Select the check box.
Appealable	Select the check box.
Members Only for Contact Log	Leave blank.
Ownership Strategy	Leave blank.

### Configuring evidence types for cases

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

### Procedure

1. From the **Administration Workspace** tab, expand the **Shortcuts** menu, and click **Case > Integrated Cases**.
2. From the integrated case, select the **Evidence Types** tab and select **Add Evidence**.
3. From the **Add Existing Evidence Type** window, enter the following fields that are described in the table.

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

Name	Description
Evidence Type	This setting allows the administrator to select the pre-configured evidence type to be associated with the case type.
Category	<p>This setting allows the administrator to select an evidence category to which the evidence type belongs. For example, the <i>Earned Income</i> evidence type belongs to the 'Income' evidence category.</p> <p>The evidence categories available for selection are associated with the <i>EvidenceCategory.ctx</i> code table. A new evidence category can be added to this code table and published as part of system administration.</p> <p>To provide flexibility, the same evidence type can belong to multiple categories. For example, the <i>Earned Income</i> evidence type can belong to the <i>Income</i> evidence category and the <i>Household</i> evidence category. These evidence categories are displayed within the evidence workspace and are used to define the subset of evidence types that can be captured on a case.</p> <p>Each category is available as a filter option when a caseworker captures new evidence for that evidence type within the evidence workspace. For example, for a caseworker that is capturing new evidence, they can filter the Category drop-down list by <b>All</b>. The system automatically displays all evidence types that are associated with any category within administration.</p>

Name	Description
Sort Order	This setting is used to dictate the order in which the evidence type is listed within the evidence workspace. For example, the sort order for the <i>Pension Fund</i> evidence type is set to 1 and the sort order for the <i>Real Estate</i> evidence type is set to 2. For a caseworker who views the evidence types for a category to create new evidence within the evidence workspace. The <i>Pension Fund</i> evidence type is listed first and the <i>Real Estate</i> evidence type is listed second. If no sort order is specified, the evidence types are listed in alphabetical order. By default, evidence types are listed in alphabetical order.
Preferred	This setting is used to dictate if the evidence type is to be available as a <i>Preferred</i> filter option in the evidence workspace. For example, if this setting is enabled for the <i>Earned Income</i> evidence type, the <i>Earned Income</i> evidence type is returned when a caseworker selects the <i>Preferred</i> category filter option for when new evidence is captured.

#### 4. Select **Save**.

#### Evidence types configured for Adult Social Care

Specify the following evidence types to include on the Adult Social Care integrated case. Enable all the evidence types for brokering the evidence.

Evidence Type	Category	Sort Order	Preferred	Evidence Broker Enabled
Practitioner Certificate	Medical	0	No	Yes
ASC Diagnosis	Medical	0	No	Yes
Incapacity Benefit Product Type	Medical	0	No	Yes

#### *Enabling evidence sharing on evidence types*

Evidence sharing allows evidence to be shared between agencies. Administrators can choose which evidence types to share and then to enable or disable each evidence type for sharing on each case type. For example, an integrated case type might have several evidence types and only a few of these evidence types might be suitable for sharing. Therefore, only these types are enabled. In order for sharing evidence to work at the case level, in addition to enabling evidence sharing on evidence types, agencies must be licensed for the Evidence Broker function.

#### **Procedure**

1. Select the **Administration Workspace** tab, expand the **Shortcuts** menu, and click **Case > Integrated Cases**.
2. Select the integrated case type.
3. From the **Evidence Types** tab, select **Enable** for the evidence types you want to enable sharing.

#### **Configuring a product delivery case**

Complete the following steps in the wizard to create a product delivery case.

#### **About this task**

These entities are populated as part of the product creation:

- CreoleProduct
- Product
- ProductCategory
- ProductDeliveryPattern
- ProductDeliveryPatternInfo
- ProductProvision
- ProvisionLocation

### Procedure

1. Select the **Administration Workspace** tab, expand the **Shortcuts** menu, and click **Case > Product Delivery Cases**.
2. Click **New Product...**
3. From the **Benefit** window, specify the benefit details and click **Next**. The benefit details are stored in the Product entity.

Name	Description
Name	As part of product creation, the name entry is automatically written to the <i>CT_ProductName.ctx</i> code table.
Type	As part of product creation, the type entry is automatically written to the <i>CT_ProductType.ctx</i> code table.
Eligible Participant Types	At least 1 'Eligible Participant Type' must be selected. The participant is specified in the ProductConcernRole entity.

4. From the **Categories** window, select the categories that you want to link the product delivery case to and click **Next**.

The category details are stored in the ProductCategory entity.

5. From the **Delivery Pattern** window, specify the following and click **Next**. The delivery pattern details are stored in the ProductDeliveryPatternInfo and ProductDeliveryPattern entities.

Name	Description
<b>From Date</b>	The date the delivery pattern record is valid from.
<b>Default Pattern</b>	<p>Select the check box to use the default delivery pattern.</p> <p>A default delivery pattern can be specified on the product. If an instance of the product is created automatically, the default pattern can be specified as an input. An example of where it happens in Curam is when the benefit overpayment liability product is created on foot of a reassessment.</p> <p>The default delivery pattern governs the setting of the <i>frequencyOverrideInd</i> on the product delivery that is created. If the default delivery pattern is not- zero the <i>frequencyOverrideInd</i> flag is set to false, otherwise it is set to true.</p> <p>A record exists in the Product entity under the <i>defaultProductDelPatternID</i> attribute when the default pattern is selected.</p>
<b>Delivery Method</b>	The method of delivery is stored in the MethodDelivery entity. The code that is associated with the delivery method type is defined in the <i>CT_MethodOfDelivery.ctx</i> code table.

Name	Description
<b>Offset</b>	The offset amount.
<b>Maximum Amount</b>	The maximum amount payable for the delivery frequency. If a payment exceeds the maximum amount to be issued, the payment is suspended and requires supervisor authorization.
<b>Cover Pattern</b>	The cover period for this record for example, Issue in advance. The cover patterns are defined in the <i>CT_ProductCoverPeriod.ctx</i> code table.
<b>Delivery Frequency</b>	The delivery frequency for example, recur every 1 weeks on Monday.

6. From the **Provision**, specify the following and click **Next**. The provision details are stored in the ProductProvision entity.

Name	Description
Product Provider	You can search and select the product provider.
Start Date	The start date of the product that is offered by the product provider.
Payment Method	The preferred method of payment for the Provision. for example, <i>Cash</i> .
Estimated Cost	The estimated cost of the product that is offered by the product provider.
Currency	The preferred currency for payment for the provision. The currencies are defined in the <i>CT_Currency.ctx</i> code table.
End Date	End date of the product that is offered by the product provider.
Payment Frequency	The payment frequency for the provision.

7. From the **Location**, specify the following and click **Next**. The ProvisionLocation entity stores the provision location details. It has a foreign key for the location details that are stored in the ProviderLocation entity.

Name	Description
Location	Product can be provided at many Locations and a Location can provide many Products (many-to-many relationship).
Cost	The estimated cost of a product that is offered by a product provider at a location.
Start Date	Start date of the product that is offered by a product provider at a location.
End Date	End date of a product that is offered by a product provider at a location.

8. From the **Eligibility Determination**, specify the following and click **Next**. The CreoleProduct entity stores the eligibility determination details.

Name	Description
Allow Open Ended Cases	Whether this product allows its product delivery cases (and thus its decisions and financial components) to be open-ended.
Decision Summary Display Strategy	The type of strategy to use to display a summary line for an interval in a determination for a delivery of this product. The type codes are defined in the <i>CT_DetIntsummarizerStrategy.ctx</i> code table.
Determination Comparison Strategy	The type of strategy to use to compare two determinations for a delivery of this product. The type codes are defined in the <i>CT_DeterminationCompStrategy.ctx</i> code table.

Name	Description
Reassessment Strategy	The type of strategy, which governs whether a case for this product is automatically reassessed when circumstances change. The type codes are defined in the <i>CT_ProductReassessmentStrat.ctx</i> code table.
Name	By default the <i>&lt;product-name&gt;EligibilityAndEntitlementRuleSet</i> name is displayed. The default name of the rule set can be edited within the product wizard. The rule set name must comply with the rule set naming standards as defined within the rule set schema.
Display Name	By default the <i>&lt;product-name&gt; Eligibility And Entitlement Rule Set</i> display name is displayed. The display name of the rule set can be edited within the product wizard. It can be used to record a translatable name for the rule set.

9. From the **Summary**, review the benefit product and click **Finish**.

*Product delivery case values for Adult Social Care*

The product delivery case values consist of benefit, categories, delivery pattern, provision, location, and eligibility determinations details for Adult Social Care.

Table 36: Benefit

Name	Value
Name	Incapacity Benefit
Type	Incapacity Benefit
Eligible Participant Types	Person

Table 37: Categories

Description
Social Care

Table 38: Delivery Pattern

Name	Values
<b>From Date</b>	Current date
<b>Default Pattern</b>	Select the check box.
<b>Delivery Method</b>	Cash
<b>Offset</b>	0
<b>Maximum Amount</b>	\$100,000.00
<b>Cover Pattern</b>	Issue in advance
<b>Delivery Frequency</b>	Recur every 1 week(s) on Monday

Table 39: Provision

Name	Value
Product Provider	Midway Health Insurance
Start Date	Current date
Payment Method	Cash

Name	Value
Estimated Cost	\$0.00
Currency	US Dollar
End Date	Leave the date as blank.
Payment Frequency	Recur every 1 week(s) on Monday

Table 40: Location

Name	Value
Location	Midway Office
Cost	Leave as the defaulted cost.
Start Date	Current date
End Date	Leave the date as blank.

Table 41: Eligibility Determination

Name	Value
Allow Open Ended Cases	Select the check box.
Decision Summary Display Strategy	Total weekly monetary entitlement
Determination Comparison Strategy	Compare all user-facing data
Reassessment Strategy	Automatically reassess all cases
Name	IncapacityBenefitRuleSet
Display Name	IncapacityBenefitRuleSet

### Configuring decision categories

After the product is created through the product wizard, you can apply extra configuration to explain how eligibility and entitlement results are determined. These decision pages provide a concise explanation of how a determination was reached on a case.

### Procedure

1. Select the **Administration Workspace** tab, expand the **Shortcuts** menu, and click **Case > Product Delivery Cases**.
2. Select a product delivery case.
3. Select the **Rule Set** tab, select the **Product Period** tab, and select **Copy for Edit**.
4. On the **Confirm Copy of Rules for Edit** window, select **Yes**.
5. Select the **Display Categories** tab and select **New Display Category**.

Name	Description
Name	Specify the name of the page that explains the eligibility results.
Display Order	Specify the left to right order in which the page displays the information for a category.
Display Page	Specify the name of the generated .jsp page that is displayed for the category.

Name	Description
Category Reference	Specify the user entered reference number that represents this display category.

6. Select **Save**.

**Decision categories for Adult Social Care**

Two decision categories are available for Adult Social Care. The first gives a summary of the eligibility and entitlement determinations. The second shows a comparison between the last two determinations that are run against the case.

Table 42: Incapacity Benefit Summary Category

Name	Values
Name	PD Incapacity Benefit Summary Category
Display Order	1
Display Page	PDIncapacityBenefit_summary
Category Reference	ibSummaryCategory

Table 43: Incapacity Benefit Decision Comparison Category

Name	Values
Name	Incapacity Benefit Decision Comparison
Display Order	2
Display Page	PDIncapacityBenefit_decisionComparison
Category Reference	ibCompareCategory

**Saving your product configuration files**

You must save your configurations.

**Quick reference for saving product configuration files**

Reference table for all files that are associated with saving your product configuration files.

The following descriptions detail each column of the reference table and what their purpose is.

- **Task Name:** The name of the current task.
- **Data Contained In:** The relevant file or files in the `%CURAM_DIR%\EJBServer\build\dataextractor` folder, which contains data that you need to extract for the step. `%CURAM_DIR%` is the Cúram installation directory, which by default is `C:\Merative\Curam\Development`.
- **Files To Be Copied From Data Extractor:** A list of files that the user must copy from the `%CURAM_DIR%\EJBServer\build\dataextractor` folder for each task.
- **Files To Be Created:** A list of files that need to be created by the user for each task.

Table 44: Product Configuration File Reference Table

For a summary of the table, see the table description in the previous list.

Task Name	Data Contained In	Files To Be Copied From Data Extractor	Files To Be Created
Saving the integrated case configuration Files	<i>CODETABLEITEM.dmx</i>	<i>ADMININTEGRATEDCASE.dmx</i> <i>ADMINICEVIDENCETYPEDEFLINK.dmx</i>	<i>ProductCategory.ctx</i> <i>CaseCatTypeCode.ctx</i>
Saving your product delivery configuration files		<i>APPRESOURCE.dmx</i> <i>CREOLEPRODUCT.dmx</i> <i>CREOLEPRODUCTPERIOD.dmx</i> <i>CREOLERULECLASSLINK.dmx</i> <i>CREOLERULESET.dmx</i> <i>CREOLERULESETCATEGORYLINK.dmx</i> <i>LOCALIZABLETEXT.dmx</i> <i>PRODUCT.dmx</i> <i>PRODUCTCATEGORY.dmx</i> <i>PRODUCTCONCERNROLE.dmx</i> <i>PRODUCTDELIVERYPATTERN.dmx</i> <i>PRODUCTDELIVERYPATTERNINFO.dmx</i> <i>PRODUCTPROVISION.dmx</i> <i>PROVISIONLOCAITON.dmx</i> <i>TEXTTRANSLATION.dmx</i>	<i>IncapacityBenefitRuleSet.xml</i> <i>IncapactiyBenefitDisplayRuleSe</i>
	<i>CODETABLEITEM.dmx</i>	<i>CT_ProductName.ctx</i> <i>CT_ProductType.ctx</i> <i>CT_ProductNameApprovals.ctx</i>	
Saving the decision categories configuration files		<i>CREOLEPRODUCTPERIODDISPCAT.dmx</i> <i>CREOLEPRODUCTDECISIONDISPCAT.dmx</i>	

### Saving the integrated case configuration files

The integrated case and associated dynamic evidence types are stored in the ADMININTEGRATEDCASE and ADMINICEVIDENCETYPEPEDEFLINK entities. Extract the DMX files from the database.

### Procedure

1. From the `%CURAM_DIR%\EJBServer\build\dataextractor` directory, copy the following DMX files.

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

File name	Description
<i>ADMININTEGRATEDCASE.dmx</i>	This entity stores the integrated case-specific information.
<i>ADMINICEVIDENCETYPEPEDEFLINK.dmx</i>	This entity defines a link between Integrated Case types and Dynamic Evidence types.

2. From the `%CURAM_DIR%\EJBServer\build\dataextractor\CODETABLEITEM.dmx` file, search for *ProductCatetegory* and *CaseCatTypeCode*.
3. If there is no pre-existing code table in the `%CURAM_DIR%\EJBServer\components\COMPONENT_NAME\codetable` directory, create a new code table for the following code tables. Add the content from the *CODETABLEITEM.dmx* file. If the code table exists, add the content to the existing code table.

File name	Description
<i>CT_ProductCategory.ctx</i>	This code table includes the codes to indicate the type of integrated case.
<i>CT_CaseCatTypeCode.ctx</i>	This code table includes the case categories type codes.

4. From %CURAM\_DIR%\EJBServer, run **build ctgen database**.
5. From %CURAM\_DIR%\webclient, run **build client**.
6. Save your configuration files.

### Integrated case configuration files for Adult Social Care

The Adult Social Care integrated case configurations files are in the following location in the accelerator.

Table 45: DMX files

File name	Location
<i>ADMININTEGRATEDCASE.dmx</i>	%CURAM_DIR%\EJBServer\components\AdultSocialCare\data\initial
<i>ADMINICEVIDENCETYPEDEFINITIONS.dmx</i>	%CURAM_DIR%\EJBServer\components\AdultSocialCare\data\demo

Table 46: Code table files

File name	Location
<i>CT_ProductCategory.ctx</i>	%CURAM_DIR%\EJBServer\components\AdultSocialCare\codetable
<i>CT_CaseCatTypeCode.ctx</i>	

### Saving your product delivery configuration files

The following DMX files are required to be extracted from the database and source controlled.

#### Procedure

1. From the %CURAM\_DIR%\EJBServer\build\dataextractor directory, copy the following DMX files. %CURAM\_DIR% is the Cúram installation directory, which by default is C:\Merative\Curam\Development.

Table 47: DMX files created from the product delivery wizard

File name	Description
<i>APPRESOURCE.dmx</i>	Contains product delivery application resources.
<i>CREOLEPRODUCT.dmx</i>	The CREOLE-specific details for a CREOLE-enabled product. The presence of CREOLE-specific details for a product means that the product is CREOLE-enabled; the absence of such details means that the product is a 'classic' product.
<i>CREOLEPRODUCTPERIOD.dmx</i>	A period within the lifetime of a CREOLE-enabled product. This period can have configurations (for example, decision rules, decision details rules, and other key events rules) which differ from other periods of time on the product.

File name	Description
<i>CREOLERULECLASSLINK .dmx</i>	An entity class in a CREOLE rule set. This entity holds the link between the CREOLEPRODUCTPERIOD and the CREOLERULESET entities. It links together the rule classes that define both the product structure and decision rule class.
<i>CREOLERULESET .dmx</i>	<p>A rule set is created for you through the product delivery wizard. A rule set for Cúram Express Rules. A set of rules comply with the CREOLE schema and validation constraints. The full XML definition of the rule set is held in the <i>ruleSetDefinition</i> column, but the Cúram application denormalizes certain structural details of the rule set to these entities:</p> <ul style="list-style-type: none"> <li>• Rule set dependency</li> <li>• Rule class</li> <li>• Rule class inheritance</li> <li>• Rule attribute</li> <li>• Rule attribute availability</li> <li>• Rule attribute inheritance</li> </ul>
<i>CREOLERULESETCATEGORY .dmx</i>	<p>An association between a CREOLE rule set or a CREOLE rule set edit action and a CREOLE rule set category.</p> <p>The CREOLE rule set category is set to <i>Eligibility and Entitlement: Determination</i> category when created through the product delivery wizard.</p> <p>A search by rule set category type is available through the administration. From the <b>Administration Workspace</b>, select <b>Rules and Evidence &gt; Cúram Express Rule Sets</b>, from the <b>Category</b> drop-down list select a category that you want to search by and click <b>Search</b>.</p>
<i>LOCALIZABLETEXT .dmx</i>	<p>This entity holds a link to the text translations for each piece of text.</p> <p>The <i>CREOLEPRODUCTPERIOD.nameID</i> and the <i>CREOLERULESET.descriptionID</i> contain localizable text.</p>
<i>PRODUCT .dmx</i>	<p>An offering that is provided by the organization to participants.</p> <p>A product might originate from an external product provider, which the organization administers on their behalf and offers it to participants as if it were an organization-owned product. The product name and product type exists in the <i>CT_ProductName.ctx</i> and <i>CT_ProductType.ctx</i> code tables.</p>
<i>PRODUCTCATEGORY .dmx</i>	A category of products. The product category code is stored in the <i>CT_ProductCategory.ctx</i> code table. All Products can be placed into one or more Categories.
<i>PRODUCTCONCERNROLE .dmx</i>	The association of a product with a participant role type for which the product is applicable. There can be more than 1 participant role type that is associated with any 1 product.
<i>PRODUCTDELIVERYPATTERN .dmx</i>	The product delivery pattern groups that are defined for a product. The product delivery pattern details are stored in the <i>PRODUCTDELIVERYPATTERNINFO</i> entity.
<i>PRODUCTDELIVERYPATTERN .dmx</i>	The product delivery pattern information is defined in a product delivery pattern group, associating a delivery method with a product, with their respective offset.
<i>PRODUCTPROVISION .dmx</i>	An association between product and product provider.

File name	Description
<i>PROVISIONLOCATION.dmx</i>	A product that is provided at a location. It implements the facts that a product can be provided at many locations and a location can provide many products (many-to-many relationship).
<i>TEXTTRANSLATION.dmx</i>	Stores the text translations for various locales. This entity does not require optimistic locking as the locking is controlled by the parent <i>LOCALIZABLETEXT</i> entity.  The <i>CREOLEPRODUCTPERIOD.nameID</i> and the <i>CREOLERULESET.descriptionID</i> contain text.

- From the `%CURAM_DIR%\EJBServer\build\dataextractor\CODETABLEITEM.dmx` file, search for *ProductName*, *ProductType*, and *ProductNameApprovals*.
- If there is no pre-existing code table in the `%CURAM_DIR%\EJBServer\components\COMPONENT_NAME\codetable` directory, create a new code table for the following code tables. Add the content from the *CODETABLEITEM.dmx* file. If the code table exists, add the content to the existing code table.

Table 48: Code tables files

File name	Description
<i>CT_ProductName.ct</i>	As part of product creation, the name entry is automatically written, to the <i>ProductName</i> code table.
<i>CT_ProductType.ct</i>	As part of product creation, the type entry is automatically written, to the <i>ProductType</i> code table.
<i>CT_ProductNameApprovals.ct</i>	As part of product creation, the name entry is automatically written, to the <i>ProductNameApprovals</i> code table. The code table is related to setting up of case approvals. It holds all the entries in the <i>ProductName</i> code table, but the reason it's separate is because there is an extra entry in there for 'All Products'.

- From the `%CURAM_DIR%\EJBServer\build\dataextractor` directory, copy the following blob and clob file from the generated extracted directory and place it in your custom component.

Table 49: blob / clob files created

File name	Description
<i>AppResource_&lt;number&gt;</i>	Contains the product wizard default eligibility entitlement rule set labels, which are used in the default rule set as annotations.
<i>CREOLERuleSet_&lt;number&gt;</i>	Contains the rule set, which is generated by the dynamic product wizard.
<i>TextTranslation_&lt;number&gt;</i>	Contains the text for the <i>Name</i> and the <i>Eligibility/Entitlement Rule Set</i> , which is displayed on the product period.

- From `%CURAM_DIR%\EJBServer`, run **build ctgen database**.
- From `%CURAM_DIR%\webclient`, run **build client**.
- Save your configuration files.

*Product delivery configuration files for Adult Social Care*

The Adult Social Care product delivery case configurations files are in the following location in the accelerator.

Table 50: DMX files created from the product delivery wizard

File name	Description	Location
<i>APPRESOURCE.dmx</i>	For Adult Social Care, the dynamic UIM pages are the application resources that are specified.	%CURAM_DIR%\EJBServer\components\IncapacityBenefit\data\initial
<i>CREOLEPRODUCT.dmx</i>	<p>For Adult Social Care, the following are set for the CREOLE-specific details:</p> <ul style="list-style-type: none"> <li>• Allows the product delivery case decisions and financial components to be open-ended.</li> <li>• The determination comparison strategy is set to <i>Compare all user-facing data</i>. The strategy type is used to compare two determinations for a delivery of this product.</li> <li>• The determinations interval summary strategy is set to <i>Total weekly monetary entitlement</i>. This type of strategy that is used to display a summary line for an interval in a determination for a delivery of this product.</li> </ul>	<div style="border: 1px solid blue; padding: 5px;"> <p><b>Note:</b> %CURAM_DIR% is the Cúram installation directory, which by default is <i>C:\Merative\Curam\Development</i>.</p> </div>

File name	Description	Location
-----------	-------------	----------

CREOLEPRODUCTPERIOD

From the product delivery case, select **Rule Sets > Product Periods > tabs**, the following are set for the CREOLE product period for Adult Social Care:

- *CREOLEPRODUCTPERIOD* stores the start for the period and end date if any specified.
- The product delivery wizard creates the following default descriptions, which are displayed on the **Product Periods** tab:

Name	Description
Name	The default name that is created through the wizard is <i>Custom product period</i> . The description name is stored in the <i>TEXTTRANSLATION.dmx</i> file and linked through the <i>LOCALIZABLETEXT.dmx</i> .
Eligibility/Entitlement Rule Set	<b>The rule set description is defaulted to</b> <ProductName> <i>Eligibility And Entitlement Rule Set</i> . The <ProductName> is stored in the <i>CT_ProductName.ctx</i> code table. The <i>CREOLERULESET.descriptionID</i> stores the identifier and description is stored in the <i>TEXTTRANSLATION.dmx</i> file, which is linked through the <i>LOCALIZABLETEXT.dmx</i> .
Eligibility/Entitlement Rule	<b>The rule is specified as follows through annotations</b> <ProductName> <i>Case</i> .
Product Structure Rule Set:	The <b>Product Structure Rule Set</b> is set the same way as the <b>Eligibility/Entitlement Rule Set</b> .
Product Structure Rule:	<b>The product rule is specified as follows through annotations</b> <ProductName> <i>Product</i> .

There are 3 links to the rule class (*CREOLERULECLASSLINK.dmx*) from the *CREOLEPRODUCTPERIOD.dmx* file:

Name	Description
productStructureRCLID	The RCLID <i>productStructureRCLID</i>

File name	Description	Location
<i>CREOLERULECLASSLIST</i>	<p>For Adult Social Care, the following rule classes are specified:</p> <ul style="list-style-type: none"> <li>The primary product <i>IncapacityBenefitProduct</i>, the primary product structure rule class <i>IncapacityBenefitProductCase</i>, and the primary product program recommendations <i>IncapacityBenefitProgramRecommender</i> for the Incapacity Benefit product in the <i>IncapacityBenefitRuleSet.xml</i> rule set.</li> <li>The <i>IncapacityBenefitSummaryCategory</i> rule class is linked to the <i>IncapacityBenefitDisplayRuleSet.xml</i> rule set.</li> </ul>	
<i>CREOLERULESET.dmx</i>	<p>The following are the 3 rules classes that are defined for Adult Social Care. Each rule class references an associated rule set definition XML file. The rules built file is reference instead of the actual file.</p> <ul style="list-style-type: none"> <li><i>IncapacityBenefitRuleSet.xml</i></li> <li><i>IncapacityBenefitDisplayRuleSet.xml</i></li> </ul>	
<i>CREOLERULESETCATEGORY</i>	<p>For Adult Social Care, the <i>IncapacityBenefitRuleSet</i> is set to <i>Eligibility &amp; Entitlement: Determination</i> category and the <i>IncapacityBenefitDisplayRuleSet</i> is set to the <i>Eligibility &amp; Entitlement: Display</i> category.</p>	
<i>LOCALIZABLETEXT.dmx</i>	<p>This entity holds a link to the text translations for each piece of text. The <i>CREOLEPRODUCTPERIOD.nameID</i> and the <i>CREOLERULESET.descriptionID</i> contain localizable text.</p>	
<i>PRODUCT.dmx</i>	<p>For Adult Social Care, the default configurations are stored which are the same for which is set through the product delivery wizard. The only differences are as follows:</p> <ul style="list-style-type: none"> <li>The product name and type are specified in the <i>CT_ProductName.ctx</i> and <i>CT_ProductType.ctx</i> code tables.</li> </ul>	
<i>PRODUCTCATEGORY.dmx</i>	<p>For Adult Social Care, the product category is <i>Social Care</i>.</p>	
<i>PRODUCTCONCERNROLE</i>	<p>For Adult Social Care, the <i>Person</i> concern role type.</p>	

File name	Description	Location
<i>PRODUCTDELIVERYPATTERNINFO</i>	The product delivery pattern groups that are defined for a product. The product delivery pattern details are stored in the <i>PRODUCTDELIVERYPATTERNINFO</i> entity.	%CURAM_DIR%\EJBServer\components\IncapacityBenefit\data\demo
<i>PRODUCTDELIVERYPATTERN</i>	For Adult Social Care, delivery pattern name is set to <i>Weekly By Cash</i> and the maximum amount is set to <i>100000.00</i> .  The product delivery wizard sets the delivery pattern name to <i>Weekly by EFT</i> and the maximum amount is set to <i>0.00</i> .	
<i>PRODUCTPROVISION</i>	For Adult Social Care, the product is linked to the <i>Midway Health Insurance</i> product provider.	
<i>PROVISIONLOCATION</i>	For Adult Social Care, the provider location is set to <i>Midway Office</i> . It also links to the product provision entity, that establishes an association to product provider.	
<i>TEXTTRANSLATION</i>	Stores the text translations for various locales. This entity does not require optimistic locking as the locking is controlled by the parent <i>LOCALIZABLETEXT</i> entity.  The <i>CREOLEPRODUCTPERIOD.nameID</i> and the <i>CREOLERULESET.descriptionID</i> contain text.	%CURAM_DIR%\EJBServer\components\IncapacityBenefit\data\initial

Table 51: Rule sets specified in the *CREOLERULESET.dmx* file.

File name	Description	Location
<i>IncapacityBenefit</i>	Containing the product delivery eligibility entitlement rules.	%CURAM_DIR%\EJBServer\components\IncapacityBenefit\CREOLE_Rule_Sets
<i>IncapacityBenefit</i>	Containing the determinations for display time. It displays to the user an eligibility result for the coverage period within the case.	

Table 52: Code tables files

File name	Location
<i>CT_ProductName.ctx</i>	%CURAM_DIR%\EJBServer\components\AdultSocialCare\codetable
<i>CT_ProductType.ctx</i>	
<i>CT_ProductNameApproval</i>	Does not exist in Adult Social Care.

**Product delivery case property change for Adult Social Care**

If demonstration data is used as part of the Incapacity benefit product delivery, the following property must be set to YES. Demonstration data must not be used in any production environments.

File name	Property name	Value	Description	Location
Application	curam.participant.creatende	YES	<p>The participant data case is created when you register a user (if the curam.participant.creatende is enabled) and is enabled by default.</p> <p>For demonstration users like James Smith, the case is only created if the property is set to <i>Yes</i>.</p> <p>This property is only to cater for demonstration data but is not intended for use in production environment.</p>	<p>%CURAM_DIR%\EJBServer\components\IncapacityBenefit\properties</p> <div style="border: 1px solid blue; padding: 5px; margin-top: 10px;"> <p><b>Note:</b> %CURAM_DIR% is the Cúram installation directory, which by default is C:\Merative\Curam\Development.</p> </div>

**Saving the decision categories configuration files**

The decision categories are configured and stored in the CREOLEPRODUCTPERIODDISPCAT and CREOLEPRODUCTDECISIONDISPCAT entities. The following DMX files are required to be extracted from the database and source controlled.

**Procedure**

1. From the %CURAM\_DIR%\EJBServer\build\dataextractor directory, copy the following DMX files.

**Note:** %CURAM\_DIR% is the Cúram installation directory, which by default is C:\Merative\Curam\Development.

File name	Description
CREOLEPRODUCTPERIODDISPCAT.dmx	<p>Display categories and display category decision details are stored on the CreoleProductDecisionDispCat and CreoleProductPeriodDispCat entities. Entries exist in the CreoleProductPeriodDispCat to link the CreoleProductPeriod, CreoleProductPeriodDispCat, and CreoleRuleClassLink entities.</p>

File name	Description
CREOLEPRODUCTDECISIONDISPCAT.dmx	<p>Greater Product Decision DispCat contains the categories of display rules for product delivery eligibility. Each category entry corresponds to a tab on the decision display page. The category defines a display order for each tab on the page.</p> <p>The display page name for product delivery eligibility is also specified, PDIncapacityBenefit_summary. The categoryRef attribute defines the field that is passed from the dynamic UIM to the program recommendation display rules infrastructure.</p> <p>The display page (UIM page and its related .properties file) must be defined within the data/clob folder. The location of the display UIM page and its related .properties files are defined in the AppResource entity.</p>

2. From %CURAM\_DIR%\EJBServer, run **build database**.
3. Save your configuration files.

### Decision category files for Adult Social Care

The Adult Social Care decision rules configuration files are in the following location in the accelerator.

Table 53: DMX files associated with the Adult Social Care decision categories

File name	Location
CREOLEPRODUCTPERIODDISPCAT.dmx	%CURAM_DIR%\EJBServer\components
CREOLEPRODUCTDECISIONDISPCAT.dmx	\IncapacityBenefit\data\initial

**Note:** %CURAM\_DIR% is the Cúram installation directory, which by default is C:\Merative\Cúram\Development.

Table 54: Dynamic UIM pages

File name	Location
PDIncapacityBenefit_summary.uim and associated properties file.	%CURAM_DIR%\EJBServer\components
PDIncapacityBenefit_diagnosisSubscreen.uim and associated properties file.	\IncapacityBenefit\data\initial\clob
PDIncapacityBenefit_summaryDropDownPane.uim and associated properties file.	\ProductDelivery
PDIncapacityBenefit_decisionComparison.uim and associated properties file.	

## Creating the IEG files for the intake process

The following cover the file that is required for creating the intake process. The files include the IEG script, schema, and domains definition for storage.

### Quick reference for specifying IEG files to create and store

Reference table for all files that are associated with saving your IEG files.

The following descriptions detail each column of the reference table and what their purpose is.

- **Task Name:** The name of the current task.
- **Data Contained In:** The relevant file or files in the `%CURAM_DIR%\EJBServer\build\dataextractor` folder, which contains data that you need to extract for the step. `%CURAM_DIR%` is the Cúram installation directory, which by default is `C:\Merative\Curam\Development`.
- **Files To Be Copied From Data Extractor:** A list of files that the user must copy from the `%CURAM_DIR%\EJBServer\build\dataextractor` folder for each task.
- **Files To Be Created:** A list of files that need to be created by the user for each task.

Table 55: IEG File Reference Table

Task Name	Data Contained In	Files To Be Copied From Data Extractor	Files To Be Created
Specifying the IEG files to create and store			ASCApplication.xml ASCApplication.properties ASCApplication.xsd ASCApplicationDomains.xsd IEGSCRIPTINFO.dmx DATASTORESCHEMA.dmx APPRESOURCE.dmx

### Specifying the IEG files to create and store

The intake process provides the facility for the user to enter the information to be stored within the data store. The files that are created are the IEG script, the schema, and domains specific for the fields that are defined in the IEG script.

### Procedure

1. Create the IEG script, the schema, and the domains files.

File name	Description
<code>&lt;IEGScriptName&gt;</code>	<p>The IEG script consists of various element references to configure such as questions, list-questions, help-text, summary-page to name just a few. It also includes meta-display elements, which provide information on how elements to be displayed. The IEG script can be configured so that display elements are configured with a logical flow. The flow-control elements include loops, validations, and conditions.</p> <p>The IEG script also consists of structural elements to allow for grouping of sections. The ability to create Java callouts to start code</p>

File name	Description
	that is not part of IEG to run operations for the script execution. You also can split a large IEG script up in to maintainable sub scripts.
<code>&lt;ScriptName&gt;.xsd</code>	The <i>Cúram Data Store</i> entity and entity fields are defined within the <code>&lt;ScriptName&gt;.xsd</code> file.
<code>&lt;ScriptName&gt;Domains.xsd</code>	The domains for each of the fields that are referenced in the schema are defined within the <code>&lt;ScriptName&gt;Domains.xsd</code> file.

2. Create the following DMX files to store the reference to the IEG script, the schema, and the domains files created.

File name	Description
<code>IEGSCRIPTINFO.dmx</code>	The script provides the facility for the user to enter the information to be stored within the data store. Each script's location is defined in <code>IEGSCRIPTINFO.dmx</code> file.
<code>DATASTORESHEMA.dmx</code>	The location of the <code>&lt;Script&gt;.xsd</code> and <code>&lt;Script&gt;Domains.xsd</code> are defined within the <code>DataStoreSchema.dmx</code> .
<code>APPRESOURCE.dmx</code>	Images ( <code>.png</code> files) and text descriptions ( <code>.properties</code> files) used by the IEG script must be defined and their locations that are referenced in the <code>AppResource.dmx</code> .

3. From `%CURAM_DIR%\EJBServer`, run **build database**.
4. Save your configuration files.

### IEG files created for Adult Social Care

The following are the IEG-related files configured for Adult Social Care.

Table 56: DMX files

File name	Description	Location
<code>IEGSCRIPTINFO.dmx</code>	This file includes the reference to the <code>ASCApplication.xml</code> IEG script file.	<code>%CURAM_DIR%\EJBServer\components\AdultSocialCare\data\initial</code>
<code>DATASTORESHEMA.dmx</code>	Includes the references to the <code>ASCApplicationDomains.xsd</code> and <code>ASCApplication.xsd</code> schema files.	<div style="border: 1px solid blue; padding: 5px;"> <b>Note:</b> <code>%CURAM_DIR%</code> is the Cúram installation directory, which by default is <code>C:\Merative\Curam\Development</code>.                 </div>
<code>APPRESOURCE.dmx</code>	Includes the reference to multiple image files and IEG script properties files specific for Adult Social Care.	

Table 57: IEG script files

File name	Description	Location
<code>ASCAApplication</code>	<p>The script that is written for Adult Social Care was relatively small, when compared to the large-scale scripts. It contains the following sections:</p> <ul style="list-style-type: none"> <li>• About The Claimant</li> <li>• Medical</li> <li>• Finish</li> </ul> <p>Some of the questions and answers in the <b>About The Claimant</b> section are not required for determining eligibility and entitlement. But were included to give the script that 'real world' example. The <b>About The Claimant</b> answers map to the <i>ASC Household Member</i>, <i>ASC Living Arrangement</i>, and <i>ASC Alien</i> entities, which are part of the Common Evidence component.</p>	<code>%CURAM_DIR%\EJBServer\components \ASCIntakeConfiguration\data\initial \clob</code>
<code>ASCAApplication</code>	<p>This property specifies the text that is displayed for each section and the finish and quit page identifiers are specified.</p>	<code>%CURAM_DIR%\EJBServer\components \ASCIntakeConfiguration\data\initial \blob</code>

Table 58: Schema files

File name	Description	Location
ASCAApplicationSchema.xsd	<p>The Adult Social Care application schema shows how to map the data that is captured as part of the intake process to the <i>Cúram Data Store</i> (CDS) entity. In particular the mapping to the parent-child relationship, the <i>PractitionerCertificate</i> (parent), <i>ASCDiagnosis</i> (child) relationship.</p> <p>The schema definition of the <i>PractitionerCertificate</i> CDS entity shows that there is a 0..n relationship <code>minOccurs="0"</code> <code>maxOccurs="unbounded"</code> between <i>PractitionerCertificate</i> and <i>ASCDiagnosis</i>.</p> <p>The schema shows that the <i>Person</i> (CDS) entity maps to the CDS entities of <i>ResidentialAddress</i>, <i>MailingAddress</i>, and the <i>PractitionerCertificate</i>.</p>	%CURAM_DIR%\EJBServer\components \ASCIntakeConfiguration\data\initial \clob
ASCAApplicationDomains.xsd	<p>The domains for each of the fields that are referenced in the Adult Social Care application schema are defined within the <code>&lt;ScriptName&gt;Domains.xsd</code> file.</p>	

## Configuring the intake process

Complete the following tasks to configure the intake process to capture data about a client when they apply to an organization for benefits or services.

### **Configuring the intake configurations**

Complete the following administration tasks to configure the intake application.

#### **Configuring evidence type**

From the application case, you can add evidence types to be available on the application case. To add evidence to an application case, only evidence, which is configured against that application case type is available.

#### **Procedure**

1. Select the **Administration Workspace** tab, expand the **Shortcuts** menu, and select **Universal Access > Application Cases** and select the application case.
2. From the **Evidence Type** tab, click **Add Evidence Type**.

Name	Description
Evidence Type	The dynamic evidence type name. The evidence types are defined in the <i>CT_EvidenceType.ctx</i> code table.
Category	The evidence category for the related evidence type in the context of the case type. The category is defined in the <i>CT_EvidenceCategory.ctx</i> code table. The available values: <i>All, Preferred, Partipant Data, Hosehold, Income, Resources, Expenses, or Assets.</i>
Sort Order	The sort order is a numeric value that is used to order evidence types. If multiple types have the exact sort order value, the alphabetic ordering of the type code is used.
Preferred	The preferred indicator means that the evidence type is displayed in the Preferred Evidence list for the case the evidence is being added to. It makes it easier for a user to find and add more commonly used evidence types.

### 3. Click **Save**.

#### *Evidence type values for Adult Social Care*

Specify the following values for creating the evidence types, which are captured and mapped as part of the Adult Social Care intake application process.

Name	Value
Evidence Type	ASC Alien
Category	Household

Name	Value
Evidence Type	ASC Household Member
Category	Household

Name	Value
Evidence Type	ASC Living Arrangement
Category	Household

Name	Value
Evidence Type	ASCDiagnosis
Category	Medical

Name	Value
Evidence Type	Incapacity Benefit Product Type

Name	Value
Category	Medical

Name	Value
Evidence Type	Practitioner Certificate
Category	Medical

### Configuring a basic program

A client can apply for a program, which is delivered by an organization. The program is used for the intake and processing the application. A remote system can be configured if required.

### Procedure

1. Select the **Administration Workspace** tab, expand the **Shortcuts** menu, and select **Universal Access > Programs** and click **New Program...**
2. From the **New Program** window, specify the following values for creating a program.

Name	Description
Name	Specify the localizable name for the program.
Intake System	Specify the system that is used for the intake processing.
Application Case	Specify the name of the case configuration record that is linked to the program. The case configuration record is used with the case type attribute to determines the type of case that is used to process an application for this program.
Case Processing System	The case system is configured in the Cúram WorkspaceServices <i>CT_ProcessingSystemType.ctx</i> code table.
Online Integrated Case Type	The online integrated case type is configured in the <i>CT_ApplicationType.ctx</i> code table.
Integrated Case Strategy	As a result of a successful program authorization, a product delivery is created and associated with either a new integrated case or an existing integrated case. This field is used to dictate the behavior.  The integrated case strategy is configured in the Cúram WorkspaceServices <i>CT_IntegratedCaseStrategy.ctx</i> code table. The available values: <i>New, Existing (Exact Client Match), Existing (Exact Client Match) or New, or Existing (Any Client Match) or New.</i>
Client Selection Strategy	The strategy to be used when a client is added from the application case to the product delivery created as a result of authorization of this program.  The client selection strategy is configured in the Cúram WorkspaceServices component

Name	Description
	<i>CT_ClientSelectionStrategy.ctx</i> code table. The available values: <i>All Clients, Rules, or User Selection</i> . The client selection strategy is only applicable when the Program Authorization Strategy of the application case is 'Program'.
Submit Product Delivery	The submit product delivery indicator, indicates whether the product delivery created in respect of this program is automatically submitted for approval.
Product Delivery Type	The product delivery type is defined in the <i>CT_PROGRAMTYPE.ctx</i> code table.

### 3. Click **Save**.

#### *Program values for Adult Social Care*

Specify the following values for configuring and incapacity benefit program for Adult Social Care.

Name	Value
Name	Adult Social Care Application Case
Intake System	Curam
Application Case	Adult Social Care Application Case
Case Processing System	Curam
Online Integrated Case Type	Adult Social Care
Integrated Case Strategy	New
Client Selection Strategy	All Clients
Submit Product Delivery	True
Product Delivery Type	Incapacity Benefit

### **Configuring the internal application**

The internal application represents an application form that is used to provide intake information for one or more programs that are offered by the social organization. The internal application references the IEG script for collecting data and the schema for the IEG script. Within the internal application, specifies the configurations for mapping the data that is stored in the Cúram Data Store to the evidence entities.

### **Procedure**

1. Select the **Administration Workspace** tab, expand the **Shortcuts** menu, and select **Universal Access > Applications**.
2. Select the **Internal Application** tab and click **New Internal Application...**
3. From the **New Internal Application** window, specify the following fields:

Name	Description
Name	The localizable name for the application.
Question Script	A reference to the IEG script that is used to capture the data for the application.
Schema	The name of the data store schema that is used by the intake script to capture data.
Submit on Completion Only	This indicator determines whether the caseworker can submit the application before you complete the intake script. If it is set to false, the caseworker is presented with the option to send the application to the organization at any point from which they exit the intake script. Otherwise, the caseworker can submit the application to the organization on completion of the entire script.

4. Click **Save**.
5. Expand the internal application and from the action menu, click **Add Program**.
6. Select the application case that is created from the drop-down list and click **Save**.
7. Select the **Programs** tab and from the action menu, click **Create Mapping**.
8. Specify an effective date and click **Save**.
9. Expand the program to see the newly created mapping with the status of In Edit.
10. From the action menu, click **Release** to change the status of the data mapping to Released.

#### *Internal application values for Adult Social Care*

Specify the following for configuring an internal application for Adult Social Care.

Name	Value
Name	Adult Social Care Internal Application
Question Script	ASCApplication
Schema	ASCApplication
Submit on Completion Only	True

#### **Optionally configure the register as prospect property value**

When new clients are added to the application they are automatically registered as prospects. The Adult Social Care application registers new clients as persons instead. Changing this default behavior is done by modifying the value of the Register as Prospect Property system value from *YES* to *NO*

#### **Procedure**

1. Log into the application as a **System Administrator**.
2. Select the **System Configurations** tab, expand the **Application Data** menu, and select **Property Administration**.
3. Type *register* into the search criteria bar.
4. From the *Register as Prospect Person* property, click the *action button* to the right.
5. Click *Edit Value* from the list of options.
6. Specify the following field:

Name	Value
Name	NO

7. Click **Save**

**Saving your intake configurations files**

The files are specified for saving the intake configurations.

**Quick reference for saving intake configuration files**

Reference table for all files that are associated with saving your intake configuration files.

The following descriptions detail each column of the reference table and what their purpose is.

- **Task Name:** The name of the current task.
- **Data Contained In:** The relevant file or files in the `%CURAM_DIR%\EJBServer\build\dataextractor` folder, which contains data that you need to extract for the step. `%CURAM_DIR%` is the Cúram installation directory, which by default is `C:\Merative\Curam\Development`.
- **Files To Be Copied From Data Extractor:** A list of files that the user must copy from the `%CURAM_DIR%\EJBServer\build\dataextractor` folder for each task.
- **Files To Be Created:** A list of files that need to be created by the user for each task.

Table 59: Intake Configuration File Reference Table

Task Name	Data Contained In	Files To Be Copied From Data Extractor	Files To Be Created
Saving your application case configuration files		<code>APPLICATIONCASEADMIN.dmx</code> <code>LOCALIZABLETEXT.dmx</code> <code>TEXTTRANSLATION.dmx</code>	
Saving your evidence type configuration files		<code>CASECONFIGEVIDENCELINK.dmx</code> <code>EVIDENCETYPEVERSIONDEF.dmx</code> <code>SECURITYGROUP.dmx</code> <code>SECURITYIDENTIFIER.dmx</code> <code>SECURITYGROUPSID.dmx</code> <code>SECURITYROLEGROUP.dmx</code>	<code>EVIDENCETYPEPEDEF.dmx</code>
Saving your program configuration files		<code>PROGRAMTYPE.dmx</code> <code>LOCALIZABLETEXT.dmx</code> <code>TEXTTRANSLATION.dmx</code>	
Saving your internal application configurations source		<code>INTAKEAPPLICATIONTYPE.dmx</code> <code>TEXTTRANSLATION.dmx</code> <code>LOCALIZABLETEXT.dmx</code>	
Saving your internal application program and mapping files		<code>APPLICATIONTYPEPROGRAMLINK.dmx</code> <code>EVIDENCEMAPPINGCONFIGURATION.dmx</code> <code>APPTYPEPROGRAMVMAPPINGLINK.dmx</code> <code>EvidenceMappingConfiguration_1_6.xml</code> <code>EvidenceMappingConfiguration_1_7.xml</code> <code>EvidenceMappingConfiguration_1_8.xml</code>	
Saving the register as a prospect person property change			<code>Application.prx</code>

### Saving your application case configurations files

An application case is used to process an application for a program. The APPLICATIONCASEADMIN entity is used to store the configuration details of an application case. When the intake application is submitted, an application case is created. The following DMX files are required to be extracted from the database. Choose the entries specific to your application case as these DMX files are merged with other components.

#### Procedure

1. From the %CURAM\_DIR%\EJBServer\build\dataextractor directory, copy the following DMX files.

**Note:** %CURAM\_DIR% is the Cúram installation directory, which by default is C:\Merative\Cúram\Development.

File name	Description
APPLICATIONCASEADMIN	Stores the details, which dictate the behavior of the application case.
LOCALIZABLETEXT.dmx	The text that appears in the Cúram application and is configured through the administration application is localizable into multiple languages. This entity holds a link to the text translation for each piece of text.  The following are foreign key attributes that are equal to each other:  The localizableTextID attribute is equal to the TEXTTRANSLATION.localizableTextID.
TEXTTRANSLATION.dmx	This entity holds the text translation for various locales.  The following are foreign key attributes that are equal to each other:  The textTranslationID is equal to the APPLICATIONCASEADMIN.nameTextID.

2. From %CURAM\_DIR%\EJBServer, run **build database**.
3. Save the configuration files.

**Application case files created for Adult Social Care**

The Adult Social Care application case configurations files are in the following location in the accelerator.

File name	Location
<i>APPLICATIONCASEADMIN.CURAM_DIR%\EJBServer\components\ASCIntakeConfiguration</i>	
<i>LOCALIZABLETEXT.dmx</i>	<i>\data\initial</i>
<i>TEXTTRANSLATION.dmx</i>	<b>Note:</b> <i>%CURAM_DIR%</i> is the Cúram installation directory, which by default is <i>C:\Merative\Curam\Development</i> .

**Saving your evidence type configurations files**

The evidence is linked to a case. The CASECONFIGEVIDENCELINK entity is used to store the evidence types, which are configured and therefore available on a case. To add evidence to a case, only evidence, which is configured against that case type is available. The following DMX files are required to be extracted from the database and source controlled.

**Procedure**

1. From the *%CURAM\_DIR%\EJBServer\build\dataextractor* directory, copy the following DMX files.

**Note:** *%CURAM\_DIR%* is the Cúram installation directory, which by default is *C:\Merative\Curam\Development*.

File name	Description
<i>CASECONFIGEVIDENCELINK.dmx</i>	The evidence types are added to this DMX file.  The following are foreign key attributes that are equal to each other:  The <i>caseConfigEvidenceLinkID</i> attribute is equal to the <i>APPLICATIONCASEADMIN.applicationCaseAdminID</i>  The <i>caseConfigurationID</i> attribute is equal to the program type <i>caseConfigurationID</i> .  The <i>evidenceTypeID</i> is equal to <i>EVIDENCERULESETDEF.evidenceTypeDefID</i> .
<i>EVIDENCETYPEDEF.dmx</i>	This entity is the link point, which allows products and integrated cases to use the dynamic evidence type. It does not contain the actual dynamic evidence metadata information but its child entity EvidenceTypeVersionDef does.
<i>EVIDENCETYPEVERSIONDEF</i>	This entity defines a dynamic evidence type, which is effective from the effective from date to the date the next evidence type version in the set is effective from.

2. Set up the security authorization for each of the evidence type definitions.

File name	Description
<i>SECURITYGROUP.dmx</i>	A security group allows an administrator to group the large number of security identifiers in to a smaller number of manageable groups. A user who has a security role is a member of the security group. They have access to all the objects represented by the security identifiers that belongs to the security group.
<i>SECURITYIDENTIFIER.dmx</i>	An object to which access can be granted.
<i>SECURITYGROUPSID.dmx</i>	A link between a security group and a security identifier.
<i>SECURITYROLEGROUP.dmx</i>	A link between a security role and a security group.

3. Update the *EVIDENCETYPEVERSIONDEF.dmx* file, *xmlMetadata* attribute for each record to point to its own XML file that stores the metadata for the dynamic evidence type.
4. From `%CURAM_DIR%\EJBServer`, run **build database**.
5. Save the configuration files.

**Evidence type files created for Adult Social Care**

The following are the evidence type DMX and XML files, which are configured for Adult Social Care. The location column specifies the directory structure for where the Adult Social Care configuration files exist.

Table 60: Evidence types configuration

File name	Location
<i>CASECONFIGEVIDENCETYPEVERSIONDEF.dmx</i>	<code>%CURAM_DIR%\EJBServer\components\ASCIntakeConfiguration\data\initial</code>
<i>EVIDENCETYPEDEF.dmx</i>	<code>%CURAM_DIR%\EJBServer\components\AdultSocialCare\data\demo</code>
<i>EVIDENCETYPEVERSIONDEF.dmx</i>	

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

Table 61: Security authorization for each of the evidence type definitions

File name	Description	Location
<i>SECURITYGROUP.dmx</i>	The <i>EVIDENCETYPEDEF.securityGroupName</i> attribute is equal to the <i>SECURITYGROUP.securityGroupName</i> .	<code>%CURAM_DIR%\EJBServer\components\AdultSocialCare\data\demo</code>
<i>SECURITYIDENTIFIER.dmx</i>	There are three security identifier records created for each dynamic evidence type ( <i>IBPRODUCTTYPE</i> , <i>PRACTITIONERCERTIFICATE</i> , and <i>DIAGNOSIS</i> ). The security identifier records created are for the <i>create</i> , <i>update</i> , and <i>modify</i> actions.	

File name	Description	Location
<i>SECURITYGROUPSIDENTIFIERS</i>	<p>For Adult Social Care, the security identifiers are specified for both a generic security group and for each individual evidence type security group. Restrictions can be made on certain actions for users that are assigned to individual evidence type security groups.</p> <p>Security identifiers are linked to dynamic evidence type security groups (<i>IBPRODUCTTYPE</i>, <i>PRACTITIONERCERTIFICATE</i>, and <i>DIAGNOSIS</i>) and also for the generic evidence group (<i>EVIDENCEGROUP</i>).</p> <p>For example, a particular user's role is assigned to the <i>IBPRODUCTTYPE</i> security group. This user is allowed to <i>create</i>, <i>update</i>, and <i>modify</i> the <i>IBPRODUCTTYPE</i> evidence type. The user is restricted from performing actions on the <i>PRACTITIONERCERTIFICATE</i> and <i>DIAGNOSIS</i> evidence types due to insufficient access rights.</p>	
<i>SECURITYROLEGROUPS</i>	<p>The eligibility worker that is created for Adult Social Care is assigned to the <i>ELIGIBILITYROLE</i> role name, which is linked to the generic <i>EVIDENCEGROUP</i> group name. The user has no restrictions on any of the evidence group actions.</p>	

Table 62: Metadata for the dynamic evidence type

Name	Description	Location
<i>Metadata_Incapable</i>	Stores the XML metadata for a Dynamic Evidence Type. This file is referenced by the <i>xmlMetadata</i> attribute in the <i>EVIDENCETYPEVERSIONDEF.dmx</i> file.	%CURAM_DIR%\EJBServer\components\AdultSocialCare\data\initial\clob
<i>Metadata_Practitioner</i>	Stores the XML metadata for a Dynamic Evidence Type. This file is referenced by the <i>xmlMetadata</i> attribute in the <i>EVIDENCETYPEVERSIONDEF.dmx</i> file.	
<i>Metadata_Diagnosis</i>	Stores the XML metadata for a Dynamic Evidence Type. This file is referenced by the <i>xmlMetadata</i> attribute in the <i>EVIDENCETYPEVERSIONDEF.dmx</i> file.	

**Saving your program configuration files**

The type of program is configured and stored in the PROGRAMTYPE entity. This entity dictates the behavior of programs both on the Citizen workspace and the internal worker application. The following DMX files are required to be extracted from the database and source controlled.

**Procedure**

1. From the %CURAM\_DIR%\EJBServer\build\dataextractor directory, copy the following DMX files.

**Note:** %CURAM\_DIR% is the Cúram installation directory, which by default is C:\Merative\Curam\Development.

File name	Description
<i>PROGRAMTYPE.dmx</i>	The program type <i>caseConfigurationID</i> attribute is equal to the <i>APPLICATIONCASEADMIN.applicationCaseAdminID</i> .
<i>LOCALIZABLETEXT.dmx</i>	<p>The text that is displayed for the program is configured through the administration application. This entity holds a link to the text translation for each piece of text.</p> <p>The following are foreign key attributes that are equal to each other:</p> <p>The <i>PROGRAMTYPE.dmx</i> file includes localizable text for the following attributes <i>nameTextID</i>, <i>descriptionTextID</i>, <i>summaryTextID</i>, <i>localOfficeTextID</i>, and <i>internalDescTextID</i>. It also includes entries for the text translation for each piece of text.</p>
<i>TEXTTRANSLATION.dmx</i>	<p>This entity holds the text translation for various locales.</p> <p>The following are foreign key attributes that are equal to each other:</p> <p>The <i>localizableTextID</i> is equal to the <i>PROGRAMTYPE.nameTextID</i>.</p>

2. From %CURAM\_DIR%\EJBServer, run **build database**.
3. Save the configuration files.

### Program files created for Adult Social Care

The following are the program DMX files, which are configured for Adult Social Care.

File name	Location
<i>PROGRAMTYPE.dmx</i>	%CURAM_DIR%\EJBServer\components\ASCIntakeConfiguration\data
<i>LOCALIZABLETEXT.dmx</i>	\initial
<i>TEXTTRANSLATION</i>	<div style="border: 1px solid black; background-color: #e6f2ff; padding: 5px; display: inline-block;"> <p><b>Note:</b> %CURAM_DIR% is the Cúram installation directory, which by default is C:\Merative\Curam\Development.</p> </div>

### Saving your internal application configurations source

The internal application represents an application form that is used to provide intake information for one or more programs that are offered by the social organization. The internal application type is configured through the Cúram administration and stored in the INTAKEAPPLICATIONTYPE entity. In the sub section, the associated program, and data mappings files are specified.

**Procedure**

1. From the %CURAM\_DIR%\EJBServer\build\dataextractor directory, copy the following DMX files.

**Note:** %CURAM\_DIR% is the Cúram installation directory, which by default is C:\Merative\Curam\Development.

File name	Description
INTAKEAPPLICATIONTYPE	<p>This file stores the internal application configurations.</p> <p>The following are foreign key attributes that are equal to each other:</p> <ul style="list-style-type: none"> <li>• The application channel that is specified in the DMX file is the OOTB workspace services code table <i>CT_ApplicationChannel.ctx</i></li> <li>• The <i>intakeScriptID</i> reference the <i>IEGSCRIPTINFO.dmx scriptID</i>. See <a href="#">Specifying the IEG files to create and store on page 101</a>.</li> </ul>
TEXTTRANSLATION.dmx	<p>The <i>INTAKEAPPLICATIONTYPE.dmx</i> file includes text translation text, which is configured in the administration application for the internal application attributes <i>nameTextID</i>, <i>descriptionTextID</i>, <i>summaryTextID</i>, <i>submissionConfirmPageTitleID</i>, and <i>submissionConfirmPageTextID</i>. It also includes entries for the text translation for each piece of text.</p>
LOCALIZABLETEXT.dmx	<p>This file includes the localizable record for the internal application <i>nameTextID</i>.</p>

2. From %CURAM\_DIR%\EJBServer, run **build database**.
3. Save the configuration files.

**Internal application files created for Adult Social Care**

The files that are specified are configured when the Adult Social Care internal application configurations were created.

File name	Location
INTAKEAPPLICATIONTYPE	%CURAM_DIR%\EJBServer\components\ASCIntakeConfiguration\data
TEXTTRANSLATION.dmx	initial
LOCALIZABLETEXT	<p><b>Note:</b> %CURAM_DIR% is the Cúram installation directory, which by default is C:\Merative\Curam\Development.</p>

**Saving your internal application program and mapping files**

Programs can be added to internal applications to represent the programs that a client can apply for using the internal application. The internal application link to the program is configured through the Cúram administration and stored in the APPLICATIONTYPEPROGRAMLINK

entity. A mapping configuration is used to map data from a data store to evidence entities. The EVIDENCEMAPPINGCONFIGURATION entity is used to store the mapping configuration for a program, which is used to map data from a data store to evidence entities. A link between a program and a mapping configuration is required. The APPTYPEPROGRAMEVMAPPINGLINK entity is used to associate a program with a mapping configuration, which is used to map data from a data store into evidence entities

**Procedure**

1. From the %CURAM\_DIR%\EJBServer\build\dataextractor directory, copy the following DMX files.

**Note:** %CURAM\_DIR% is the Cúram installation directory, which by default is C:\Merative\Curam\Development.

File name	Description
APPLICATIONTYPEPROGRAMLINK.DMX	This file links the program record to the internal application.
EVIDENCEMAPPINGCONFIGURATION.DMX	This file is used to store the mapping configuration for a program, which is used to map data from a data store to evidence entities. The DMX file includes reference to the mappings files. The status attribute is chosen from the CT_EvidenceMappingStatus.ctx code table, which exists in the workspace services component.
APPTYPEPROGRAMEVMAPPINGLINK.DMX	This file is used to associate a program with a mapping configuration, which is used to map data from a data store into evidence entities.

2. From %CURAM\_DIR%\EJBServer, run **build database**.
3. Save the configuration files.

**Program and mapping files created for Adult Social Care**

The following are the Adult Social Care DMX files used to associate a program with a mapping configuration. The configurations are used to map data from a data store into evidence entities.

File name	Location
APPLICATIONTYPEPROGRAMLINK.DMX	%CURAM_DIR%\EJBServer\components\ASCIntakeConfiguration\data\initial
EVIDENCEMAPPINGCONFIGURATION.DMX	%CURAM_DIR%\EJBServer\components\ASCIntakeConfiguration\data\initial
APPTYPEPROGRAMEVMAPPINGLINK.DMX	%CURAM_DIR%\EJBServer\components\ASCIntakeConfiguration\data\initial

**Note:** %CURAM\_DIR% is the Cúram installation directory, which by default is C:\Merative\Curam\Development.

The EVIDENCEMAPPINGCONFIGURATION.dmx file stores the mapping configuration that is used to map from a data store to evidence entity. The following are the XML files, which are configured for Adult Social Care.

File name	Description	Location
<i>EvidenceMappingConf</i>	<p>The role of MDP equates to the Medical Provider, which is what was specified in the dynamic evidence editor for the practitioner attribute. The configuration for practitioner creates a representative on the system, which is added as a case participant of type Medical Provider. The setting of Medical Provider on the practitioner attribute of <i>PractitionerCertificate</i> ensures that only case participants of this type appear in the drop-down for practitioner on the create/modify pages.</p> <p>The <i>EVIDENCEMAPPINGCONFIGURATION mappingConfigurationXML</i> attribute points to the mapping file, which includes the mapping configurations for the following evidence entities:</p> <ul style="list-style-type: none"> <li>• ASCAlien</li> <li>• HeadOfHousehold</li> <li>• ASCHouseholdMember</li> <li>• PractitionerCertificate</li> <li>• ASCDiagnosis</li> <li>• ASCLivingArrangement</li> </ul>	%CURAM_DIR%\EJBServer\components \ASCIntakeConfiguration\data\initial \clob
<i>EvidenceMappingConf</i>	<p>This file includes the XML specification for transforming data in the <i>Cúram Data Store</i> to the full set of entities. The <i>EVIDENCEMAPPINGCONFIGURATION mappingXML</i> attribute points to the data map for the <i>PractitionerCertificate</i>, <i>ASCDiagnosis</i>, and Person data store entities.</p>	

File name	Description	Location
<i>EvidenceMappingConfiguration_1_8</i>	The <i>EVIDENCEMAPPINGCONFIGURATION</i> editorData attribute points to this file. The content of this file is empty for the Adult Social Care.	

### Saving the register as prospect property change

For the Adult Social Care intake process, the application is configured to register the client as a person. The Register as Prospect Person system property dictates whether clients on an application are registered as prospects or persons.

#### Procedure

1. From the `%CURAM_DIR%\EJBServer\components\COMPONENT_NAME\properties` directory, update the *Application.prx* files to specify the following property and specify the value to *No*. Create the *Application.prx* file if it does not exist.

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

Property name	Value	Default
curam.intake.registerAsProspect	No	Yes

2. From `%CURAM_DIR%\EJBServer`, run **build database**.
3. Save the configuration files.

## Creating an Evidence Management wizard

You can use the following documentation as a basis for creating your own custom Evidence Management wizard.

### **Evidence Management wizard design**

Before you start developing Evidence Management wizards, ensure that you consider the key principles behind whether or not an Evidence Management wizard is applicable.

The key principles around when to use Evidence Management wizards for performing a change in circumstance are outlined here.

- The change involves more than one evidence type,
- Evidence updates need to happen in a sequence, and
- Conditional evidence needs to be entered based on information captured as part of the change in circumstance

If these requirements are met, then the design of the Evidence Management wizard needs to be driven out by answering the following questions:

- Which evidence types need to be updated in the wizard?

- In what order does the information need to be captured?
- Are there any conditional aspects to the capturing of the change in circumstance?
- What information needs to be displayed on each page?
- Is it possible to finish on more than one page?
- What information needs to be displayed on the summary page?
- Do rules need to be written to determine further actions that need to be taken by the caseworker to complete the change in circumstance?

### ***Evidence Management wizard implementation points***

Evidence Management wizard implementation points are highlighted here.

#### **UIM Patterns**

Where a considerable amount of data is being captured as part of the process, the use of editable lists and conditional clusters can improve the overall user experience. For example, conditional clusters can reduce the number of pages needed in the wizard.

#### **Structs to model for dynamic evidence**

For dynamic evidence maintained through the Evidence Management wizard, it's necessary to model a static equivalent of each dynamic evidence entity that is being updated. This modeling needs to be done by using IBM® Rational® Software Architect Designer. The naming convention for this struct is *<EntityName>Dtls*.

#### **Identifiers for list items**

For structs that represent a record in a list, an attribute that represents a unique identifier for that list item needs to be added to the struct. To highlight the fact that this attribute is a non-evidence data attribute, the naming convention for the attribute is *nonEv<EntityName>ListItemID*.

### ***Modeling the Evidence Management wizard***

When you complete the Evidence Management wizard design, you must use Rational® Software Architect Designer to model the facade, operations, and structs. It can also be used for specifying which fields are mandatory.

#### **Evidence Management wizard class diagrams**

Class diagrams that represent the modeling of the required structs and operations for the Adult Social Care Evidence Management wizard facade.

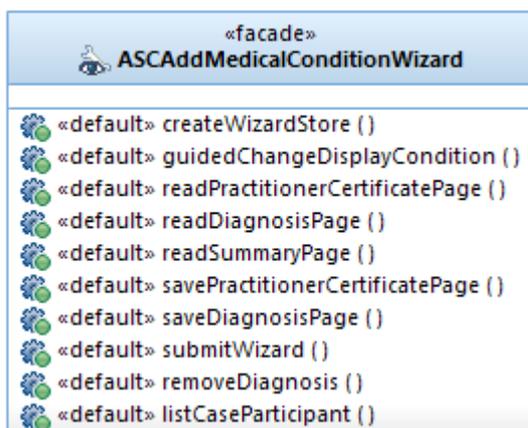


Figure 1: Evidence Management wizard facade class diagram.

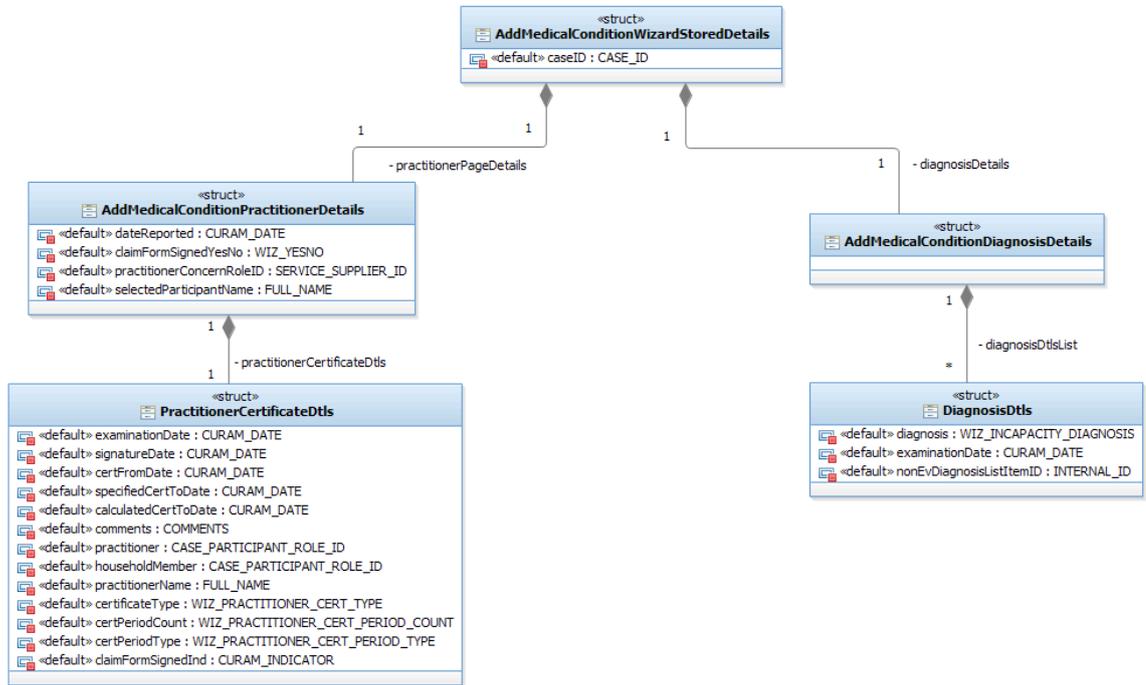


Figure 2: Evidence Management persisted storage structure class diagram.

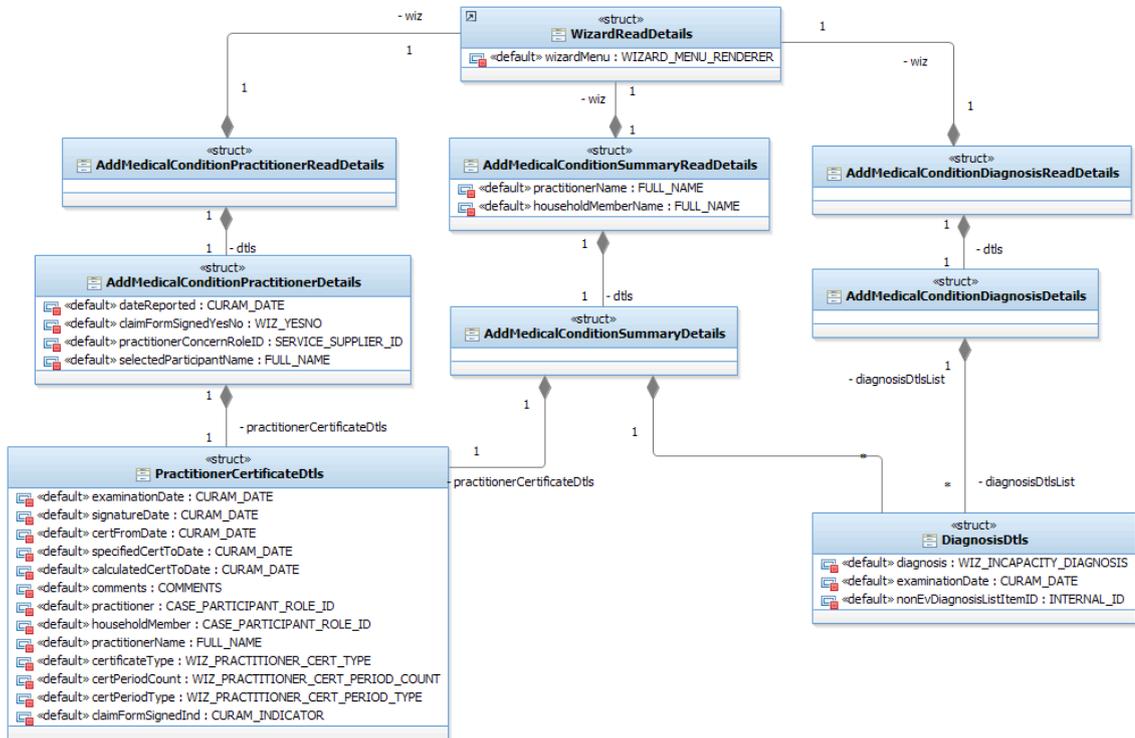


Figure 3: Evidence Management display structure class diagram.

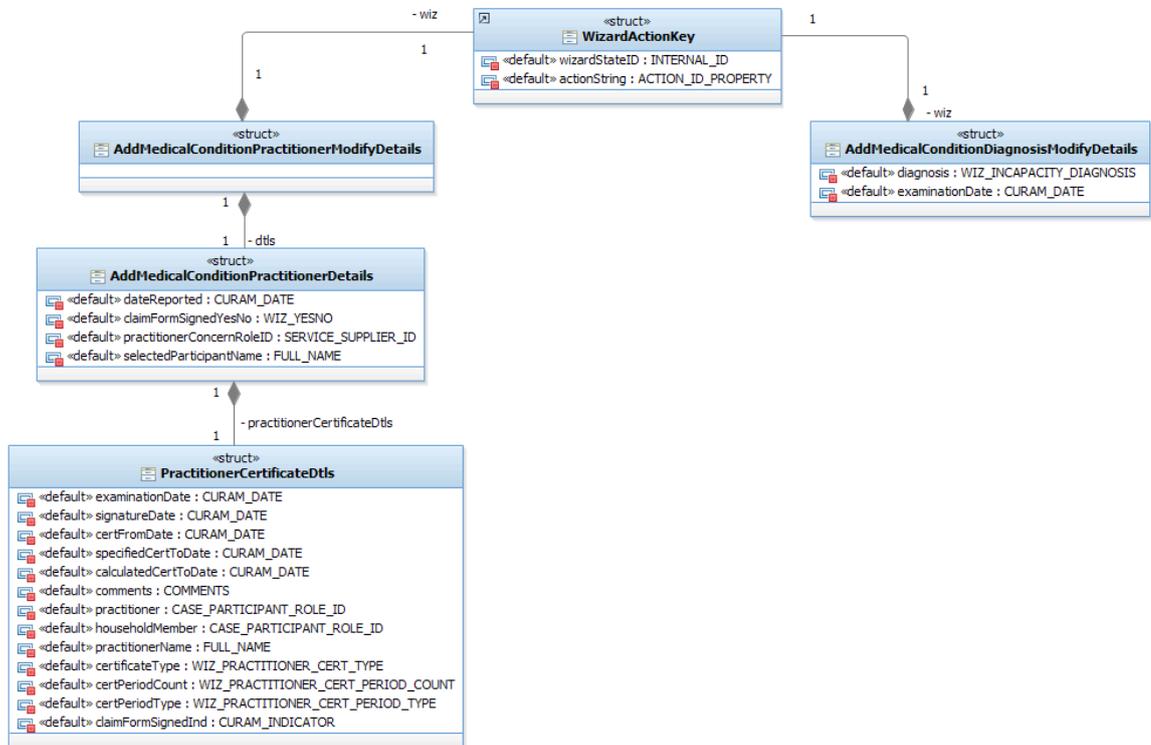


Figure 4: Evidence Management action structure class diagram.

## Modeling the Evidence Management wizard facade

When you complete the Evidence Management wizard design, you must use the Rational® Software Architect Designer to model the facade, operations, and structs.

### Before you begin

You must create the required structs.

### Procedure

1. In the model, create a facade called `ASCAddMedicalConditionWizard` with a stereotype of facade.

2. For each entry in the following table, create a new operation on the ASCAddMedicalConditionWizard facade, with the specified structs defined for input and output.

Facade Operation	Input Struct	Return Struct	Description
createWizardSetup	WizardSetupKey	WizardKey	Create the empty wizard persistent storage for the Add Medical Condition wizard.
guidedChangeDisplayCondition	CaseHeaderKey	GuidedChangeDisplayCondition	Display whether or not to display a page level action link to start the Add Medical Condition Evidence Management wizard.
listCaseParticipants	SearchCaseParticipantDetailsKey	SearchCaseParticipantDetails	Retrieve case participants by case and types.
readDiagnosisPage	WizardKey	AddMedicalConditionDiagnosisListItems	Read details for the diagnosis page of the Add Medical Condition wizard.
readPractitionerCertificatePage	WizardKey	AddMedicalConditionPractitionerDetails	Read details for the practitioner certificate page of the Add Medical Condition wizard.
readSummaryPage	WizardKey	AddMedicalConditionSummaryListItems	Read details for the summary page of the Add Medical Condition wizard.
removeDiagnosis	AddMedicalConditionDiagnosisListItemsKey		Remove a diagnosis from the persisted storage for the wizard.
saveDiagnosis	AddMedicalConditionDiagnosisModifyDetails		Save the details for the diagnosis page of the Add Medical Condition wizard.
savePractitionerCertificatePage	AddMedicalConditionPractitionerModifyDetails		Save the details for the practitioner certificate page of the Add Medical Condition wizard.

Facade Operation	Input Struct	Return Struct	Description
submitWizard	WizardActionKey	n/a	Decentralize the data persisted by the wizard, saving it across the database onto all the appropriate tables and run any configured rules to create guided changes next step actions.

### What to do next

Model the mandatory properties for each facade method.

### Modeling the mandatory properties

IBM® Rational® Software Architect Designer can be used to define which fields are mandatory in the Evidence Management wizard facade operations.

### Procedure

Model the mandatory field values according to the table for each operation that is defined in the ASCAddMedicalConditionWizard facade in the model.

Facade Operation	Mandatory Fields Value	Description
saveDiagnosisPage	diagnosis,examinationDate	The mandatory fields to successfully save the diagnosis page of the Add Medical Condition Evidence Management wizard.
savePractitionerCertificateDtls	dtls.practitionerCertificateDtls.examinationDate,dtls.pr	The mandatory fields to successfully save the practitioner certificate page of the Add Medical Condition Evidence Management wizard.

### Implementing the Evidence Management wizard

After the modeling is complete, you need to generate the Cúram model and implement the wizard functionality. Instructions on how to do that are outlined here.

### Generating the Cúram model

When you create a facade operation in IBM® Rational® Software Architect Designer, run the following build targets.

### About this task

The `build generated` target generates the code from the model, builds code tables, messages, and events but does not compile handcrafted code. If you run a `build server` target, the `compile.implemented` substep would fail as the handcrafted implementation class is not

present. Provide a dummy implementation of the modeled operations before you run the `build compile.implemented` target.

The `build database` target inserts the security identifiers that are associated with the facade operations into the database.

**Procedure**

1. Run `build` generated from `%CURAM_DIR%\EJBServer`.  
`%CURAM_DIR%` is the Cúram installation directory, which by default is `C:\Merative\Curam\Development`.
2. Refresh and build the EJBServer project inside Eclipse.
3. Open up the modeled implementation class and right click and select **Source > Override/Implement Methods** to add the stubs for the modeled operations.
4. Run `build compile.implemented database` from `%CURAM_DIR%\EJBServer`.

**Implementation files for the Adult Social Care Evidence Management wizard**

The following tables include the locations of the server-side and client-side artifacts that were created to implement the Adult Social Care Evidence Management wizard.

Table 63: Server side files for the Evidence Management implementation

Class Name	Location
ASCAddMedicalConditionW	<code>%CURAM_DIR%\EJBServer\components\AdultSocialCare\source\AdultSocialCare_src.zip\curam\ascproduct\emw\facade\impl\</code>
<div style="border: 1px solid blue; padding: 5px; background-color: #e6f2ff;"> <p><b>Note:</b> <code>%CURAM_DIR%</code> is the Cúram installation directory, which by default is <code>C:\Merative\Curam\Development</code>.</p> </div>	
ASCAddMedicalConditionU	<code>%CURAM_DIR%\EJBServer\components\AdultSocialCare\source\AdultSocialCare_src.zip\curam\ascproduct\emw\sl\impl</code>
ASCAddMedicalConditionN	<code>%CURAM_DIR%\EJBServer\components\AdultSocialCare\source\AdultSocialCare_src.zip\curam\ascproduct\emw\sl\impl</code>
ASCAddMedicalConditionE	<code>%CURAM_DIR%\EJBServer\components\AdultSocialCare\source\AdultSocialCare_src.zip\curam\ascproduct\emw\sl\mappers\impl</code>
DiagnosisEvidenceMapper	<code>%CURAM_DIR%\EJBServer\components\AdultSocialCare\source\AdultSocialCare_src.zip\curam\ascproduct\emw\sl\mappers\impl</code>
PractitionerCertificate	<code>%CURAM_DIR%\EJBServer\components\AdultSocialCare\source\AdultSocialCare_src.zip\curam\ascproduct\emw\sl\mappers\impl</code>
ASCIntegratedCaseMenuSta	<code>%CURAM_DIR%\EJBServer\components\AdultSocialCare\source\AdultSocialCare_src.zip\curam\ascproduct\tab\impl</code>
ASCIntegratedCaseNavSta	<code>%CURAM_DIR%\EJBServer\components\AdultSocialCare\source\AdultSocialCare_src.zip\curam\ascproduct\tab\impl</code>
NavigationConst.java	<code>%CURAM_DIR%\EJBServer\components\AdultSocialCare\source\AdultSocialCare_src.zip\curam\ascproduct\tab\impl</code>
Module.java	<code>%CURAM_DIR%\EJBServer\components\AdultSocialCare\source\AdultSocialCare_src.zip\curam\ascproduct\impl</code>

Class Name	Location
ASCAddMedicalConditionWizard	%CURAM_DIR%\EJBServer\components\AdultSocialCare\message
GuidedChange.xml	%CURAM_DIR%\EJBServer\components\AdultSocialCare\message
CT_GuidedChangeWizardType	%CURAM_DIR%\EJBServer\components\AdultSocialCare\codetable
CT_WizYesNo.ctx	%CURAM_DIR%\EJBServer\components\AdultSocialCare\codetable
APPRESOURCE.dmx	%CURAM_DIR%\EJBServer\components\AdultSocialCare\data\initial
ASCAddMedicalConditionWizard	%CURAM_DIR%\EJBServer\components\AdultSocialCare\data\initial\blob
DefaultIntegratedCase.na	%CURAM_DIR%\EJBServer\components\AdultSocialCare\tab\IntegratedCase
DefaultIntegratedCase.pr	%CURAM_DIR%\EJBServer\components\AdultSocialCare\tab\IntegratedCase
DefaultIntegratedCaseMer	%CURAM_DIR%\EJBServer\components\AdultSocialCare\tab\IntegratedCase
DefaultIntegratedCaseMer	%CURAM_DIR%\EJBServer\components\AdultSocialCare\tab\IntegratedCase

Table 64: Client side files for the Evidence Management implementation

File Name	Location
ASC_guidedChangeFromIntegratedCase	%CURAM_DIR%\webclient\components\AdultSocialCare\GuidedChanges
ASC_guidedChangeFromIntegratedCase	%CURAM_DIR%\webclient\components\AdultSocialCare\GuidedChanges
ASC_completedGuidedChangeFromIntegratedCase	%CURAM_DIR%\webclient\components\AdultSocialCare\GuidedChanges
ASC_completedGuidedChangeFromIntegratedCase	%CURAM_DIR%\webclient\components\AdultSocialCare\GuidedChanges
ASCAddMedicalCondition_start.ui	%CURAM_DIR%\webclient\components\AdultSocialCare\GuidedChanges\AddMedicalCondition
ASCAddMedicalCondition_start.pr	%CURAM_DIR%\webclient\components\AdultSocialCare\GuidedChanges\AddMedicalCondition
ASCAddMedicalCondition_diagnosis	%CURAM_DIR%\webclient\components\AdultSocialCare\GuidedChanges\AddMedicalCondition
ASCAddMedicalCondition_diagnosis	%CURAM_DIR%\webclient\components\AdultSocialCare\GuidedChanges\AddMedicalCondition
ASCAddMedicalCondition_removed	%CURAM_DIR%\webclient\components\AdultSocialCare\GuidedChanges\AddMedicalCondition
ASCAddMedicalCondition_removed	%CURAM_DIR%\webclient\components\AdultSocialCare\GuidedChanges\AddMedicalCondition
ASCAddMedicalCondition_summary	%CURAM_DIR%\webclient\components\AdultSocialCare\GuidedChanges\AddMedicalCondition
ASCAddMedicalCondition_summary	%CURAM_DIR%\webclient\components\AdultSocialCare\GuidedChanges\AddMedicalCondition
ASCAddMedicalCondition_resolve	%CURAM_DIR%\webclient\components\AdultSocialCare\GuidedChanges\AddMedicalCondition
ASCAddMedicalCondition_resolve	%CURAM_DIR%\webclient\components\AdultSocialCare\GuidedChanges\AddMedicalCondition
Constants.properties	%CURAM_DIR%\webclient\components\AdultSocialCare\GuidedChanges\AddMedicalCondition

## Modifying person match

Person match is a process for automatically matching applicants to registered clients. Complete the following tasks to modify the person match criteria.

### **Quick reference for saving person match configuration files**

Reference table for all files that are associated with saving your person match criteria.

The following descriptions detail each column of the reference table and what their purpose is.

- **Task Name:** The name of the current task.
- **Data Contained In:** The relevant file or files in the %CURAM\_DIR%\EJBServer\build\dataextractor folder, which contains data that you need to extract for the step. %CURAM\_DIR% is the Cúram installation directory, which by default is C:\Merative\Curam\Development.
- **Files To Be Copied From Data Extractor:** A list of files that the user must copy from the %CURAM\_DIR%\EJBServer\build\dataextractor folder for each task.
- **Files To Be Created:** A list of files that need to be created by the user for each task.

Table 65: Person Match File Reference Table

Task Name	Data Contained In	Files To Be Copied From Data Extractor	Files To Be Created
Saving your person match search configuration files	PERSONMATCHCASESEARCHCRITERIA.dmx TEXTTRANSLATION.dmx LOCALIZABLETEXT.dmx		Application.prx

### **Modifying the person match search criteria**

Based on the weight scores that are configured determines whether the person is a match or not and dictates whether clients on an application are registered as prospects or persons. The person match search criteria is used when extra clients are captured on an internal application form and client registration is not defined online.

#### **About this task**

The PERSONMATCHSEARCHCRITERIA entity is used to store the search criteria details for the Person Match. The search criteria is configured by the user in admin and stored on the PERSONMATCHSEARCHCRITERIA entity. The Adult Social Care application takes advantage of the pre-configured person match criteria that is delivered with the Cúram Platform. No additional criteria needs to be added.

#### **Procedure**

1. Log in to the Cúram system as a user with administrator credentials.
2. Select the **Administration Workspace** tab, expand the **Shortcuts** menu, and click **Participants > Person Match Search Criteria**.
3. From the **New Person Match Search Criterion** window, specify the following fields and click **New Search Criteria....**

Name	Description
Name	The name of the person search criteria.
Reference	The unique reference that is associated with each person search criteria. This reference name is entered by the user when they are adding the search criteria.
Weight	The score that is associated with a valid match based on the search criteria. The score determines whether the person is a conclusive match, an inconclusive match, or not a match, which is based on the thresholds configured.
Enabled	Indicates whether the person match search criteria is enabled.

4. Click **Save**.

### ***Saving your person match search configuration files***

Save your person match search configuration files.

#### **Procedure**

1. From the `%CURAM_DIR%\EJBServer\build\dataextractor` directory, copy the following DMX files.

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

File name	Description
<code>PERSONMATCHSEARCHCRITERIA.dmx</code>	This file is used to store the search criteria details for the person match.
<code>TEXTTRANSLATION.dmx</code>	The text name that is entered through the admin configuration is stored in this file.  The following are foreign key attributes that are equal to each other:  The <code>localizableTextID</code> is equal to the <code>PERSONMATCHSEARCHCRITERIA.nameTextID</code>
<code>LOCALIZABLETEXT.dmx</code>	This file includes the localizable text for the person match search name.  The following are foreign key attributes that are equal to each other:  The <code>localizableTextID</code> attribute is equal to the <code>TEXTTRANSLATION.localizableTextID</code> .

2. From `%CURAM_DIR%\EJBServer`, run **build database**.

3. Save the configuration files.

## **Configuring workflows**

Complete the following tasks to configure a workflow that automate a business process.

### ***Configuring a struct to enact a workflow***

Complete the following steps to configure a struct that passes parameters to a workflow to ensure that it is triggered under the correct circumstances. Modeling an enactment struct is required only if one or more parameters that are required for the enactment of a workflow is a non-integer data type.

## Before you begin

For Cúram development, you must use Rational Software Architect to configure structs.

### Creating a model

Complete the following steps to create a model for the struct.

#### Procedure

1. In the Rational® Software Architect Designer Eclipse workspace, click the **EJBServer > Models** directory, and right-click **Models > Create Model**.
2. From the **Create Model** wizard, click **Next**.
3. Click **Template Selection > Blank Package**, specify a file name and directory and click **Finish**.

#### Model values for Adult Social Care

Specify the following values for creating a model in Adult Social Care.

Table 66: Model values

File name	Directory
AdultSocialCare	%CURAM_DIR%\EJBServer\components \AdultSocialCare\model

**Note:** %CURAM\_DIR% is the Cúram installation directory, which by default is C:\Merative\Curam\Development.

### Creating a package

Complete the following steps to create a package.

#### Procedure

1. From **Windows Explorer**, go to the %CURAM\_DIR%\EJBServer\components\  
%COMPONENT\_NAME%\models folder and create the following directory *Packages*  
*Workflow* for the new .efx file.  
%CURAM\_DIR% is the Cúram installation directory, which by default is C:\Merative  
\Curam\Development.
2. In the Rational® Software Architect Designer Eclipse workspace, click **EJBServer > Models**  
> **%MODEL\_NAME%**, right-click the model, and click **Add Class > Package**.
3. Update the name of the new package.

#### Package values for Adult Social Care

Specify the following values for creating a workflow package in Adult Social Care.

Table 67: Package values

Name
Workflow
DomainDefinitions

## Creating a fragment

Complete the following steps to create a fragment class for the workflow package.

### Procedure

1. In the Rational® Software Architect Designer Eclipse workspace, right-click **EJBServer** and click **Refresh**.
2. Click **EJBServer > Models > %MODEL\_NAME% > %PACKAGE\_NAME%**.
3. Right-click the package name and click **Refactor > Create Fragment** to create a new fragment.
4. From the **Save As Fragment** window, enter a fragment name and click **OK**.

### Fragment values for Adult Social Care

Specify the following values for creating the fragments in Adult Social Care.

Table 68: Fragment values

File name
Workflow.efx
DomainDefinitions.efx

## Configuring a domain definition

Complete the following steps to create a domain definition.

### Before you begin

You must open the *Core::Core::DomainDefinitions* package so that the existing Cúram domain definitions are available.

### Procedure

1. In the Rational® Software Architect Designer Eclipse workspace, right-click on the workflow package and click **Add Class > Domain Definition**.
2. From the **Domain Definitions** window, enter the following criteria for the name and type of the domain definition. When you are searching for an existing type from the **In addition to Project Scope** window, click the **Entire Workspace** option:

Name	Description
Domain definition name	The name of the domain definition, which is used as the identifier for the underlying data type that the domain definition is associated with.
Type	Cúram provides a number of data types that provides a layer of abstraction between the developer and the underlying database, and Java types. These data types can be used to define attributes, arguments, and return values, which are mapped by the SDEJ to the associated underlying data type.

3. Click **OK > Finish**.

### Domain definition values for Adult Social Care

Specify the following values for configuring a domain definition in Adult Social Care.

Table 69: Domain definition values

Name	Type
CORRECTION_SET_ID	INTERNAL_ID

### Creating a struct

Complete the following steps to create a struct.

#### Procedure

1. In the Rational® Software Architect Designer Eclipse workspace, from the *EJBServer/models* directory, right-click the **%PACKAGE\_NAME%** > **New Class** > **Struct**.
2. Rename the struct.

### Struct values for Adult Social Care

Specify the following values for creating a struct in Adult Social Care.

Table 70: Struct values

Struct value
CaseIDCorrectionSetID

### Adding an attribute to a struct

Complete the following steps to add an attribute to a struct in Rational® Software Architect Designer. All Cúram attributes must begin with a lowercase character.

#### Procedure

1. In the Rational® Software Architect Designer Eclipse workspace, right-click the struct and click **Add Attribute** > **Default**.
2. From the **Create Default Attribute** window, enter the following values:

Name	Type
The name of the struct attribute.	The attribute type. For information on data types refer to the Cúram modeling guide.

3. Click **OK** > **Finish**.
4. Repeat these steps to add extra attributes to the struct.

*Struct attribute values for Adult Social Care*

Specify the following values for creating a struct attribute in Adult Social Care.

Table 71: Struct attribute values

Name	Type	Description
caseID	CASE_ID	You need to have the Core::Core::DomainDefinitions package open in order for the existing CASE_ID domain definition be available in your search.
correctionSetID	CORRECTION_SET_ID	You must create the CORRECTION_SET_ID domain definition for this to be available.

**Adding the code package property to a struct**

To use a struct from Eclipse, you must specify a package structure for the struct to allow it to be imported.

**Procedure**

1. In the Rational® Software Architect Designer Eclipse workspace, click on the **%PACKAGE\_NAME%** that contains the struct.
2. From the **properties** window, click **Advanced > curam.profile > Code\_Package**.
3. Enter a package structure, which is used to access the struct from Eclipse.
4. From the upper-left corner of the Rational® Software Architect Designer user interface, click **File > Save** to save any changes.
5. From the command prompt, change to the **%CURAM\_DIR%** directory and run *setEnvironment.bat*.  
**%CURAM\_DIR%** is the Cúram installation directory, which by default is *C:\Merative\Curam\Development*.
6. From the **%CURAM\_DIR%\EJBServer** directory, run the following command to generate the model changes:

```
build generated
```

*Code package values for Adult Social Care*

Specify the following code package values for a struct in Adult Social Care.

The following package structure is displayed for the struct which is imported into the Adult Social Care code:

```
curam.ascproduct.struct.CaseIDCorrectionSetID
```

Table 72: Code package value

Code_Package value
ascproduct

## **Configuring a workflow**

Complete the following tasks to configure a basic work flow.

### **Configuring a workflow event**

Complete the following steps to configure an event.

#### **About this task**

You configure an event for a manual and event wait activities. The application needs to inform the workflow engine when the work associated with a task is complete or when an event occurred.

#### **Procedure**

1. Log in to the Cúram system as a user with the administrator credentials.
2. Select the **Administration Workspace** tab, expand the **Shortcuts** menu, and click **Workflow > Events > New Event Class...**
3. From the **Create Event Class** window, enter a name in the event class field and click **Save**.
4. From the event class, select the action icon and click **New Event Type**.
5. From the **New Event Type** window, enter the name in the event type field and click **Save**.

#### *Event values for Adult Social Care*

Specify the following values for configuring two events for Adult Social Care.

*Table 73: Event class and type values*

<b>Event Class</b>	<b>Event Type</b>
ASCCertExtension	ExtendCert
ASCTask	Closed

## **Configuring a workflow**

Complete the following tasks to configure a workflow that creates a manual task when certain conditions are met.

### *Configuring a workflow process definition*

Complete the following steps to configure a workflow process definition that defines a business process flow.

#### **About this task**

A workflow process definition describes the flow of a business process in terms that are understood by the Workflow Engine.

#### **Procedure**

1. Select the **Administration Workspace** tab, expand the **Shortcuts** menu, and click **Workflow > New Process**.
2. From the **New Workflow Process** window, enter the following details.

Name	Description
<b>Name</b>	The name of the workflow. This name represents a short description of the workflow and is displayed throughout the Process Definition Tool. It is also a localizable string with no parameters.
<b>Technical ID</b>	The technical identifier allows the workflow to be referenced in code.
<b>Category</b>	The category list is specified from the <i>CT_ProcessCategory.ctx</i> code table.
<b>Start Process Split Type</b>	A split allows multiple threads of work to be done at the same time while a join is the reciprocal synchronization point for those threads.
<b>End Process Join Type</b>	Each activity has a split and a join type. The activity has only one outgoing or incoming transition a type of <code>none</code> is assigned to the split or join respectively. The other two split and join types, <code>choice</code> (also known as XOR) and <code>parallel</code> (also known as AND).
<b>Description</b>	An optional description of the workflow.

3. Click **Save**.

### Workflow values for Adult Social Care

Specify the following values for configuring a manual workflow for Adult Social Care.

Table 74: Manual workflow values

Name	Value
<b>Name</b>	ASC Extend Practitioner Certificate
<b>Technical ID</b>	ASCExtendPractitionerCertificate
<b>Category</b>	Tasks
<b>Start Process Split Type</b>	None
<b>End Process Join Type</b>	None
<b>Description</b>	The workflow creation process contains a manual activity that creates a task and assigns it to the case owner for action.

### Configuring workflow data objects

Complete the following steps to create a data object and attributes for the workflow. Workflow data objects are the mechanism for passing data around within a workflow process instance.

### About this task

You can create a workflow data object (WDO) for a workflow process definition. The WDO includes the generic details of the workflow data object itself and also the details of the attributes that are associated with the WDO.

### Procedure

1. From the workflow, select the **Data Objects** tab and click **New Data Object**.
2. From the **New Data Object** window, enter the following details.

Name	Description
Name	The name of the workflow data object. For example, specify the name in CamelCase.

Name	Description
Display Name	The display name of the workflow data object. This name represents a short description of the workflow data object and is displayed throughout the Process Definition Tool. It is also a localizable string with no parameters.
List Data Object	The list data object contains a <code>BOOLEAN</code> value, which indicates whether the specified workflow data object is a list workflow data object or not. When set to <code>true</code> , the specified workflow data object acts as a list and thus can be used to make lists of data available throughout the workflow.
Initialize Attributes	The initialize attribute contains a <code>BOOLEAN</code> value, which indicates whether the attributes associated with the workflow data object must be initialized when the workflow data object is first used. The default values that are used are the same as would be set in a <code>Cúram</code> struct.
Description	The description contains a more detailed description of the workflow data object. It is also a localizable string with no parameters.

3. Click **Save**.
4. From the data object created, select the action icon and click **New Attribute**.
5. From the **New Data Object Attribute** window, enter the following details.

Name	Description
Name	The name of the workflow data object attribute.
Display Name	The display name of the workflow data object attribute. The name represents a short description of the workflow data object attribute. It is a localizable string that does not contain any parameters.
Type	<p>Each workflow data object attribute that is defined must specify a type, which must be a valid <code>Cúram</code> base domain. When you create a workflow data object attribute in the Process Definition Tool, the type is selected from the <code>DomainType</code> code table.</p> <p>The type of a workflow data object attribute is used to ensure that the data mappings contained within a workflow process are compatible and do not cause failures at runtime.</p> <p>For example, if a business process object method parameter field was of type <code>STRING</code>, then the workflow data object attribute used to map the data into that field must also be of type <code>STRING</code>.</p>
Required For Enactment	<p>Enactment mappings represent the minimum amount of data that the workflow requires in order to be enacted. When you set the enactment flag to <code>true</code>, you must create an enactment mapping for each workflow data object attribute. Conversely, by setting this flag to <code>false</code> (the default) means that this workflow data object attribute is not required for the enactment of the associated process.</p> <p>The Process Definition Tool is used to create these enactment mappings. It does so by examining each workflow data object attribute (WDO) and creating a mapping for the WDOs that have the required at enactment flag set to <code>true</code>.</p> <p>When you select a released workflow process definition as a subflow process in a subflow activity, all of the workflow data objects, which are marked as required for enactment in the sub flow process must be mapped before that parent process definition can be released.</p> <p>For moere information, see the <i>Workflow Reference Guide</i>.</p>

Name	Description
Process Output	<p>A workflow process can be marked as a Web Service by setting a metadata value, which indicates that the process must be exposed as a Web Service. Enabling process object, allows the process to be able to participate in a Business Process Execution Language (BPEL) orchestrated process. It means that the process can be called from a BPEL process either synchronously or asynchronously.</p> <p>You also might need to map data out from a workflow process back into the BPEL process that called it. When you set the process output to <code>true</code>, this element indicates that the data from this workflow data object attribute must be passed back to the calling BPEL process when the Cúram workflow process completes. The default for this element is <code>false</code>.</p>

6. Click **Save** or **Save & New** to create a new WDO attribute.

### Data objects values for Adult Social Care

Specify the following values for configuring a workflow data object and its associated attributes for Adult Social Care.

Table 75: Data object values

Name	Value
Name	CaseIDCorrectionSetID
Display Name	Case and correction set identifiers

Table 76: Workflow data object attribute values

Name	Display Name	Type	Required for Enactment
CaseID	Case identifier	64 bit Integer Type	True
correctionSetID	Correction set identifier	64 bit Integer Type	True

### Configuring the enactment mappings

For each workflow data object with enactment flag set to true, complete the following steps to configure the enactment mappings.

### About this task

Enactment mappings define the data that the workflow requires to be executed.

### Procedure

1. From the workflow **Enactment** tab, select the action icon and click **Create Enactment Mappings > Yes**.

**Note:** If you create a WDO attribute that is marked as required for enactment, or modify an existing WDO attribute to be required for enactment, then your enactment mapping must be refreshed. Click **Refresh Enactment Mappings** to refresh your mappings.

2. From the attribute, select the action icon and click **Edit**.
3. From the **Edit Enactment Mapping** window, select the search icon.

4. From the **Select Struct Field** window, enter the struct name including the package structure. For example, *curam.core.struct.CaseKey*. Click **Search**.
5. Select the action for the attribute identifier, for example **caseID** and click **Save**.

**Enactment values for Adult Social Care**

Specify the following values for enacting the workflow data object attributes for Adult Social Care.

*Table 77: Enactment attribute values*

Enactment attribute	Type	Struct Field
CaseIDCorrectionSetID.caseID	64 bit Integer Type	curam.ascproduct.struct.CaseIDCorrectionSetID.c
CaseIDCorrectionSetID.correctionSetID	64 bit Integer Type	curam.ascproduct.struct.CaseIDCorrectionSetID.c

**Configuring a manual activity**

An activity in a workflow process definition defines a discrete piece of work that must be completed before a workflow process can progress. You can create a number of different types of activities.

**Configuring a manual activity**

Complete the following steps to create an activity for the workflow.

**Procedure**

1. From the workflow, select the **Activities** tab and click **New Activity...**
2. From the **New Activity** window, enter the following details.

Name	Description
Activity Name	The mandatory activity name must be unique within a specified work flow process definition and the name is displayed for the activity. The activity name is a localizable string, and a validation ensures that a specified activity name is unique for each locale that is specified.
Category	You can optionally put an activity into a category to facilitate searches based on activities. The category attribute has no functional effect on the activity.  The category is specified in the <i>CT_ActivityCategory.ctx</i> code table.
Type	You can select from the following activity types in the process definition tool: <ul style="list-style-type: none"> <li>• Automatic</li> <li>• Decision</li> <li>• Event Wait</li> <li>• LoopBegin</li> <li>• Manual</li> <li>• Parallel</li> <li>• Route</li> <li>• Subflow</li> </ul> For more information on each of the activity types, see the <i>Workflow Reference Guide</i> .

Name	Description
Parallel Activity Type	A parallel activity acts as a wrapper around existing activities which allows multiple tasks to different individuals at the same time.  The parallel activity type is only applicable when an activity of type Parallel is created. There are two types of parallel activity: Manual and Decision.
Split Type Join Type	A split allows multiple threads of work to be done at the same time while a join is the reciprocal synchronization point for those threads.  Each activity has a split and a join type. The activity has only one outgoing or incoming transition a type of <code>none</code> is assigned to the split or join respectively. The other two split and join types, <code>choice</code> (also known as XOR) and <code>parallel</code> (also known as AND).
Description	An optional description can help with future process definition updates.

**3. Click Save.**

**Manual activity values for Adult Social Care**

Specify the following values for configuring a manual activity for Adult Social Care.

In Cúram workflow, a manual activity is used where the business process requires human intervention. A user might be required to make decisions, supply extra data or to action tasks in the real world such as telephoning a client. The manual activity also specifies the information that is displayed to the user when notified that they must perform a task. Also, the selection of the agents to which the work is assigned.

*Table 78: Manual activity values*

Name	Value
Activity Name	ASC Extend Certificate Decision
Category	Standard
Type	Manual
Parallel Activity Type	Leave blank
Split Type	None
Join Type	None
Description	This activity creates a task for the eligibility worker to decide whether to extend the certificate for the client.

**Configuring an event wait**

Complete the following steps to configure an event wait for a manual activity.

**About this task**

An event wait provides you with a way to notify the workflow engine when the work required is completed. You can configure an event wait and associate it with a manual activity. A manual activity requires some action to be taken by a user before the workflow can be completed.

For example, you can configure an event wait to close a task. You can implement an event wait for when a caseworker applies changes to evidence, an event is raised for closing the task.

## Procedure

1. From the workflow activity, select the **Event Wait** tab and click **Assign Event**.
2. From the **Assign Event** window, click on the search icon for **Event**.
3. From the **Select Event Type** window, select the event from the drop-down list and click **Search**.
4. Click the **Select** action to select the event type.
5. From the **Assign Event** window, click on the search icon for **Event Match Data**.
6. From the **Select Workflow Data Object Attribute For Event Match Data** window, select the workflow data object from the drop-down list and click **Search**.
7. Click the **Select** action to select the attribute name.
8. Click **Save**.

### Event wait values for Adult Social Care

Specify the following values for configuring an event wait for the manual activity for Adult Social Care.

*Table 79: Event wait values*

Event Class	Event Type	Event Match Data
ASCTask	CLOSED	CaseIDCorrectionSetID.correctionSetID

### Configuring a workflow task

Complete the following steps to configure a workflow task.

#### About this task

To notify a user that they are required to do some work as part of an automated business process, a task is assigned to them. A task is a message that appears in the users inbox. This inbox specifies the work that the user is expected to do. The task can also have a list of actions that are associated with it. Actions are links to Cúram application pages where the work required to perform the task can be performed.

## Procedure

1. From the workflow activity, select the **Task Details** tab and click **Edit**.
2. From the **Edit Task Details** window, enter the following details.

Table 80: Task details

Name	Description												
Subject Text	<p>The subject text contains the details of the parameterized message that is displayed in the task subject field. When a manual activity is executed, a task is created. When a user views their tasks in the inbox, this message represents the subject of that task.</p> <p>The text of the subject can contain replaceable strings (%k), which is replaced with the associated text parameters. A text parameter is a mapping to a workflow data object attribute. Parameter k in the list replaces %k in the text string, where k is the order of the parameter in the list. %k can be repeated within the string and thus each workflow data object attribute must only be mapped once. A format for the replaceable strings can optionally be specified by placing another letter after the replaceable string, for example, %1d, where d will format the value as a date.</p> <table border="1" data-bbox="587 676 1426 963"> <thead> <tr> <th data-bbox="595 687 986 719">Formatting Letter</th> <th data-bbox="994 687 1418 719">Format As</th> </tr> </thead> <tbody> <tr> <td data-bbox="595 729 986 772">s</td> <td data-bbox="994 729 1418 772">string</td> </tr> <tr> <td data-bbox="595 783 986 825">n</td> <td data-bbox="994 783 1418 825">numeric</td> </tr> <tr> <td data-bbox="595 836 986 878">d</td> <td data-bbox="994 836 1418 878">date</td> </tr> <tr> <td data-bbox="595 889 986 932">z</td> <td data-bbox="994 889 1418 932">date/time</td> </tr> <tr> <td data-bbox="595 942 986 963">t</td> <td data-bbox="994 942 1418 963">time</td> </tr> </tbody> </table>	Formatting Letter	Format As	s	string	n	numeric	d	date	z	date/time	t	time
Formatting Letter	Format As												
s	string												
n	numeric												
d	date												
z	date/time												
t	time												
Priority	A task can optionally contain a priority and this metadata contains those details. The priority is selected in the Process Definition Tool and is taken from the CT_TaskPriority.ctx code table.												
Mapped Priority	You can dynamically evaluate the priority of the task and use workflow data objects to set this field.												
Administration SID	This field allows an administration security identifier to be specified for a manual task. This allows a user in a group that is associated with the specified security identifier to modify the task details, although the task might be reserved by another user in the application.												
Manual Forward Allowed	This is a Boolean flag that is used to indicate that if the task generated from the execution of the associated manual activity can be forwarded to another user. When a task is generated, it is allocated to an agent to carry out the work. Setting this flag to true (the default is true) allows that agent to forward that task to another agent to carry out the specified work. You are required to provide implementation behind this functionality if you want to enable it.												
Deadline Override Allowed	This is a Boolean flag that is used to indicate that if the deadline that is associated with the manual activity task can be overridden. Setting the value of this flag to true (the default is false) indicates that the deadline time can be changed after the task is created by the workflow engine. For more information about deadlines, see the <i>Workflow Reference Guide</i> .												
Initial Comment	This field allows an initial comment mapping to be specified for the manual task. The value of the workflow data object attribute that is used in this mapping is used to place a record in the TaskHistory table when the associated manual activity is executed.												

3. Click Save.

4. If you specified text parameters in the subject text, configure the text parameters by completing the following steps:
  - a) From the **Task Details** tab, click **New Text Parameter** only when you specified a text parameter in the subject text.
  - b) From the **New Text Parameter** window, select the search icon to open the **Select Workflow Data Object Attribute** window.
  - c) Select **TaskDetails** from the drop-down list and click **Search**.
  - d) Click the **Select** action to select the attribute and click **Save**.

#### Task values for Adult Social Care

Specify the following values for configuring a task for Adult Social Care.

Table 81: Task values

Name	Value
Subject Text	Practitioner Certificate needs to be extended
Priority	High

#### Configuring a task action

Complete the following steps to configure the actions on the task.

#### About this task

The actions that are specified for the manual activity task are links to Cúram application pages where the work required to perform the task can be performed. The pages that are specified in the task actions must be valid Cúram pages and must be available in the Cúram application. The parameters in the pages must match the parameters that are specified as action link parameters in the associated task actions.

#### Procedure

1. From the workflow activity, select the **Task Details** tab and select the **Actions** section from the side navigation bar and click **New Action...**
2. From the **New Action** window, enter the following details.

Name	Description
Page ID	The identifier of the target Cúram page on which a user can perform the required action.
Action Text	The description that is displayed on the action text link.
Principal Action	Actions can be defined as primary or secondary actions. Principal actions usually contain the links to the Cúram pages on which a user can perform the actual required work. Secondary actions usually contain links to supporting information that the user assigned to do the work can refer to while they work on the assigned task.

Name	Description
Open In Modal	<p>If this indicator is set to true, then the page that is specified by the action link is opened in a modal dialog.</p> <p>When the field is set to false (default), then the client infrastructure decides how to open the link in the same fashion as it does with any other link in the application. That is, if the page is part of a tab configuration, then it opens the appropriate tab. Otherwise, it replaces the action link home page in the content area of the current tab.</p>
Multiple Occurring Action	<p>This field signifies that this action represents a multiple occurring action. When the activity is executed, the workflow engine creates one action record for each item in the list workflow data object specified as the multiple occurring action.</p> <div style="border: 1px solid blue; padding: 5px; margin-top: 10px;"> <p><b>Note:</b> When the multiple occurring action is specified for an action, then an attribute from the associated list workflow data object must be used as a link parameter for the action.</p> </div>

**3. Click Save.**

Action values for Adult Social Care

Specify the following values for configuring an action for a task for Adult Social Care.

Table 82: Action values

Name	Value
Page ID	Evidence_workspaceActiveHighLevel
Action Text	Evidence Workspace Home Page
Principal Action	Yes

Configuring a business object association

Complete the following steps to configure a business object association.

**About this task**

Manual activities, and indeed workflow in general, perform operations on entities that exist in the application. For this reason, it can be useful to associate a task with the entities that are related to it for that process. Business object associations essentially provide links between a task and any application entities of interest for that process. The quintessential examples in Cúram include the Case and Concern entities.

**Procedure**

1. From the workflow activity, select the **Business Object Associations** tab and click **New Business Object Association...**
2. From the **New Business Object Association** window, enter the following details.

Name	Description
Business Object Type	The business object type is specified within the CT_BusinessObjectType.ctx code table.

Name	Description
Data Object Attribute	<p>This field specifies the business object type for the business object association for the manual activity.</p> <p>The workflow data object attribute that is mapped to the business object type for a manual activity business object association must be valid. This attribute type must be assignable to a type LONG as the data object attribute represents a mapping to a unique identifier. For example, a case identifier or participant identifier.</p>

### 3. Click **Save**.

#### Business object association values for Adult Social Care

Specify the following values for configuring a business object association for Adult Social Care.

Table 83: Business object association values

Name	Value
Business Object Type	Case
Data Object Attribute	CaseIDCorrectionSetID.caseID

#### Configuring an allocation strategy

Complete the following steps to configure an allocation strategy.

#### About this task

An organization typically has many human agents at various levels of responsibility that can perform work for a process definition. To select a specific agent or group of agents that can do the work for a specific manual activity, an allocation strategy is assigned to the activity.

There are three types of allocation strategies that are currently supported by Cúram workflow:

- **Function:** selecting a function allocation strategy results in the invocation of the specified allocation function when the associated activity is executed by the workflow engine.
- **Cúram Express rules (CER):** selecting a Cúram Express rules (CER) allocation strategy results in the specified ruleset that is associated with the activity is executed.
- **Target:** selecting an allocation strategy of type target results in an agent or group of agents is assigned to the work directly.

#### Procedure

1. From the workflow activity, select the **Allocation Strategy** tab and click **Edit**.
2. From the **Edit Allocation Strategy** window, select an allocation strategy type and select the search icon.
3. From the **Select Allocation Target** window, select the **Select** action to select the allocation target.
4. Click **Save**.

#### Allocation strategy values for Adult Social Care

Specify the following values for configuring an allocation strategy for Adult Social Care.

Table 84: Allocation strategy values

Allocation Strategy Type	Allocation Function
Function	curam.core.sl.intf.WorkflowAllocationFunction.caseOwnerAllocation

### Configuring a transition

Complete the following steps to configure a transition.

### About this task

Transitions provide the links between activities. They are the primary flow control construct and dictate the order in which activities are executed. Transitions are unidirectional and an activity can have multiple outgoing and incoming transitions. Every process definition must have one start and one end activity. A process definition can be thought of informally as a directed graph in which activities are the vertices, transitions are the arcs, and every path from the start activity eventually leads to the end activity.

### Procedure

1. From the workflow, select the **Transitions** tab and click **New Transition...**
2. From the **New Transition** window, enter the following details.

Name	Description
From Activity	This is the 64-bit identifier of the source activity of the transition.
To Activity	This is the 64-bit identifier of the target activity of the transition.

3. Click **Save** or **Save & New** to create a new transaction.

### Transition values for Adult Social Care

Specify the following values for configuring a transition for Adult Social Care.

Table 85: Transition values

From Activity	To Activity
Start Process Activity	ASC Extend Certificate Decision
ASC Extend Certificate Decision	End Process Activity

### Editing an existing workflow

Complete the following steps to edit an existing released workflow.

### Procedure

1. From the **Administration Workspace** tab, expand the **Shortcuts** menu, and click **Workflow > Released Processes**.
2. From the **Released Processes** tab, use the pagination search arrows to find the existing workflow.
3. Select the action icon for the workflow you want to edit and click **Edit > Save**.

4. From the **Shortcuts** menu, click **Workflow > In-Edit Processes** to see the list of modifiable workflow processes.
5. Select the workflow name to open the workflow in a new tab.

*Workflow created for Adult Social Care*

The following is the workflow configured for Adult Social Care.

Table 86: Workflow details

Name	Category
ASC Extend Practitioner Certificate	Tasks

**Releasing a workflow**

Complete the following step to save an in-edit workflow.

**Procedure**

From the workflow **Home** tab, click on the **Release Processes** to release the workflow.

**Saving your workflow configurations**

You must extract your workflow configurations to ensure that they are not lost when you run a `build database` target. The following include step-by-step instructions on how to save your workflow configurations.

**Saving the released workflow configurations**

Complete the following steps to save your workflow configurations.

**Procedure**

1. From the workflow **Home** tab, click on the **ACTIONS** icon, select **Export** to save the XML file.
2. Save the XML file in the `%CURAM_DIR%\EJBServer\components\  
%COMPONENT_NAME%\workflow` directory. `%CURAM_DIR%` is the Cúram installation directory, which by default is `C:\Merative\Curam\Development`.

*Workflow file configurations for Adult Social Care*

The Adult Social Care workflow is saved in the following location.

Table 87: Workflow file

File Name	Location
ASCEExtendPractitionerCertificate.xml	<code>%CURAM_DIR%\EJBServer\components\AdultSocialCare\workflow</code>

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

## Creating an event file

To save an event, you must create and save an event EVX file to include the event class and event type.

### About this task

When you extract the event configurations from the database, the configurations are stored in an *EVENTCLASS.dmx* and *EVENTTYPE.dmx* file. The contents from these files need to be added to an associated *.evx* file. Use the example code for creating an event file. There is no need to extract the configurations file from the database if you know what the event class and type.

### Procedure

1. From the `%CURAM_DIR%\EJBServer\components\%COMPONENT_NAME%\events` directory, create the *.evx* file for the event class and event type. Example code is specified here.

```
<events package="curam.events">
  <event-class identifier="ASCTask" value="ASCTask">
    <annotation>ASC Task Closure</annotation>
    <event-type identifier="Closed" value="Closed"/>
  </event-class>
</events>
```

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

2. From `%CURAM_DIR%\EJBServer`, run **build evgen database**. Your configurations file for the workflow and the event are built that uses this build target.

## Event configuration files for Adult Social Care

The Adult Social Care event files are stored in the following location.

Table 88: EVX files

File Name	Description	Location
<code>ASCTaskClosure.ev</code>	<p>For Adult Social Care, this event is used to close a task. An event is raised to close a task when a caseworker click <b>Apply Changes</b> from the evidence workspace.</p>	<p><code>%CURAM_DIR%\EJBServer\components\AdultSocialCare\events</code></p>
<code>ASCTaskCertExtension.ev</code>	<p>As part of the Adult Social Care intake process, the application script includes questions where the caseworker can specify a date for which a claimant can return to work.</p> <p>A configurable property is implemented which specifies the number of days for which an event needs to be raised to notify the case owner to extend the certification period.</p> <p>This event enacts the Adult Social Care work flow and creates a task and assigns it to the case owner. The task is displayed in the case owners inbox.</p>	<p><b>Note:</b> <code>%CURAM_DIR%</code> is the Cúram installation directory, which by default is <code>C:\Merative\Curam\Development</code>.</p>
	<p><b>Note:</b> This event is not configured as part of the work flow configurations.</p>	

## Implementing a workflow enactment

Implementing and exporting an enactment struct to enact a workflow.

### Implementing a workflow enactment

You can implement a workflow enactment in your custom code to start your workflow.

### About this task

As part of the intake process, the application script includes questions where the caseworker can specify a date for which a claimant can return to work. A configurable property is implemented which specifies the number of days for which an event needs to be raised to notify the case owner to extend the certification period. This event enacts the workflow and creates a task, which is then assigned to the case owner. The task is displayed in the case owners inbox.

## Procedure

1. Import your enactment struct into your code by using the following example code:

```
import your_package_path.your_struct_name;
```

2. The following pseudo code shows the logic that is used to extend the *PaymentInstrumentEvent* class to run custom logic every time a payment is generated on a case:

```

// A method which extends the PaymentInstrumentEvent class to check if specific
type of evidence is present and is about to expire.
// If the evidence is going to expire within a predetermined period this
function will enact the workflow which will
// generate a task for the case owner who can then extend the expiration date if
necessary.

public void checkExpirationAndEnactWorkflow(
    PaymentInstrumentDtls paymentInstrumentDtls){

    1. Get concernRoleID from paymentInstrumentDtls.
    2. Get financial instructions from case using the date ranges to search for the
appropriate payments.
    3. Retrieve the Case for the Payment using concernRoleID and searching for
payments with a status of PROCESSED. Example:
        ILIUnprocessedDtlsList iliDtlsList = instructionLineItemObj
            .searchByConcernRoleIDCategoryStatus(rmKey);
        long caseID = iliDtlsList.dtls.firstItem;
    4. Use caseID to return a list of financialComponents and look at the financial
schedule for the payments, only concentrating on the live components.

        FCStatusKey fcStatusCaseKey = getAllLiveFCInCase(caseID);
        final FinancialComponentDtlsList fcDtlsList = financialComponentObj
            .searchByStatusCaseID(fcStatusCaseKey);

    5. Loop through all live financial components to get the end date of the final
payment.
    6. If the end date of the financial schedule is less than or equal to a
predetermined period from now, 14 days in this instance, the workflow is enacted.
Example:

        int timeToEnd = endOfScheduleDate - currentDate;
        if (timeToEnd <= kExpirationPeriod) {
            7. Look for specific evidence type in active case evidence list.
            for (final ECActiveEvidenceDtls evidenceDtls : evidenceList.dtls) {
                if (evidenceDtls.evidenceType == EVIDENCE_TYPE) {
                    8. Take correctionSetID from evidence that is to be modified.
                    correctionSetID = evidenceDtls.correctionSetID;
                    break;
                }
            }
        }

    9. Enact the workflow:
        final MyEnactmentStruct enactmentStruct = new MyEnactmentStruct();

        //Enactment struct modeled for Adult Social Care requires a caseID and
CorrectionSetID for enactment.
        enactmentStruct.caseID = caseKey.caseID;
        enactmentStruct.correctionSetID = Long.parseLong(correctionSetID);

        //The enactment struct is added to the list twice here, this is because it
must be fully populated when passed for enactment,
//each enactment argument is picked up in sequence so the CaseID will be
picked up the first time the struct is passed in and the correctionSetID will be
picked up the second time.
        final List<Object> enactmentStructList = new ArrayList<Object>();
        // Add caseID to the enactment list through the struct.
        enactmentStructList.add(enactmentStruct);
        // Add correctionSetID to the enactment list through the struct.
        enactmentStructList.add(enactmentStruct);
        //Trigger the workflow
        EnactmentService.startProcess("ExtendEvidenceWorkflow",
enactmentStructList);
    }
}

```

## Implementing a post event

You can implement a post event class to inject custom code that performs a specific piece of functionality.

### About this task

The following example provides you with a scenario, which closes a task that is assigned to a caseworker once they perform a specific action.

### Procedure

1. To trigger an event when the caseworker clicks **Apply Changes** from the evidence workspace, you can extend the *EvidenceActivationEvents* class.
2. You must override the *postActivation* function inside your new event class. Inside the method, you can include custom logic, which can be run every time that the event is triggered.
3. The following pseudo code shows the logic that is used to check for a specific type of evidence is modified and closes the associated task.

```
// A function that closes the case owner's task if they have modified a specific
// evidence type on the case in question.
// This function is triggered when the case owner clicks "Apply Changes" after
// modifying the evidence.
@Override
public void postActivation(
    final CaseKey key, final EvidenceList list) {
    1. For each piece of evidence in the list {
        If this evidence is a <EVIDENCE_TYPE> {
            2. For each piece of evidence, check if it is a modified version of
            existing evidence:
                if evidence.status == SUPERSEDED{
                    3. Trigger the task close event to close the case owner's
                    task if they have modified a practitioner certificate.
                    final Event event = new Event();
                    event.eventKey = EventTaskClosure.Closed;
                    4. Pass the correctionSetID to the event to make sure the
                    task associated with the correct piece of evidence is closed.
                    final long correctionSetID = evDtls.correctionSetID;
                    event.primaryEventData = correctionSetID;
                    EventService.raiseEvent(event);
                }
            }
    }
}
```

# 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.