

# **Cúram 8.1.2**

## **Configuring Evidence Guide**



## Note

---

Before using this information and the product it supports, read the information in [Notices on page 111](#)



# Edition

---

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

© Merative US L.P. 2012, 2024

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



# Contents

---

<b>Note.....</b>	<b>iii</b>
<b>Edition.....</b>	<b>v</b>
<b>1 Configuring evidence.....</b>	<b>9</b>
1.1 Configuring evidence types.....	9
Display deleted evidence.....	9
Configuring an Evidence Dashboard Group.....	9
Enabling the end dating of previous evidence when creating evidence.....	10
1.2 Configuring dynamic evidence.....	11
Dynamic evidence types and dynamic evidence type versions.....	11
Dynamic evidence lifecycle.....	23
Evolution of dynamic evidence type metadata.....	25
Security administration.....	27
Dynamic evidence editor overview.....	28
Defining a model for a client data type version.....	34
Defining the user interface.....	65
Configuring the multiple participant evidence update.....	77
Dynamic evidence rule sets.....	77
Localizing dynamic evidence.....	93
Dynamic Evidence Configuration Extractor.....	95
Features of the Dynamic Evidence Metadata Loader.....	101
Generated artefacts.....	104
Dynamic Evidence application properties.....	107
Compliance for dynamic evidence.....	107
Source code for the sample widgets.....	108
<b>Notices.....</b>	<b>111</b>
Privacy policy.....	112
Trademarks.....	112





# 1 Configuring evidence

You can configure evidence types and then associate the evidence types with application cases, integrated cases, product delivery cases, persons and prospect persons, and outcome plans.

## 1.1 Configuring evidence types

Evidence can be captured and maintained on the person tab and at the case level. To make evidence that is captured on the person tab and the at case level available to use to caseworkers, the evidence must be configured.

You can configure dynamic and static evidence types to associate them with evidence dashboard groups and to enable the end dating of evidence.

## Display deleted evidence

Administrators can use the application properties in the administration application to configure the display of deleted evidence.

### About this task

By default, deleted evidence is not displayed. The following steps outline how to enable the display of deleted evidence in the Evidence Type list or in the Evidence Object Type tab.

### Procedure

1. Log in to Cúram as a system administrator.
2. Select **System Configurations > Shortcuts > Application Data > Property Administration**.
3. Type `Display Deleted Evidence` in the **Name** field and click **Search**.
4. Select **Edit Value**.
5. Set the value to **TRUE** and click **Save** to save your changes.
6. Click **Publish** for your changes to take effect.

**Note:** Evidences are not listed on the evidence dashboard. As a result, enabling the display of deleted evidence affects only the pages where evidence is listed. Enabling the display of deleted evidence does not affect evidence dashboard groups. For more information about evidence dashboard groups, see [Configuring an Evidence Dashboard Group on page 9](#).

## Configuring an Evidence Dashboard Group

Administrators can create an evidence dashboard group to which evidence types can be added.

When the evidence dashboard group is configured, the evidence dashboard group is displayed to a caseworker on the Evidence Dashboard. When the caseworker selects the Evidence Dashboard,

the Evidence Dashboard filters the evidence so that it displays only the evidence types that are part of that particular group. Evidence dashboard groups can be created to help users perform a specific task in relation to evidence maintenance.

For example, an evidence dashboard group can be created to add a member to an integrated case. All the evidence that is needed to complete this task can be grouped. When the group is selected on the dashboard at the case level, only the evidence that belongs to the group is displayed on the dashboard.

The administrator can also set evidence types that are required for that specific group. Evidence types that are flagged as required are used to prompt the user that the evidence type must be entered as part of the task this evidence group relates to.

Evidence dashboard groups can be configured on application cases, integrated cases, and product delivery case types.

The following list outlines the steps to configure case types:

1. Log in to Cúram as an administrator.
2. Select **Administration Workspace > Shortcuts > Cases**.
3. Select the applicable case type.
4. Click the **Evidences** tab.
5. Click **Dashboard Group**.

Evidence dashboard groups are specific to the case type against which the dashboard group is configured.

## Enabling the end dating of previous evidence when creating evidence

An administrator can configure evidence types so that the caseworker can end date existing evidence when the caseworker is creating evidence, depending on the requirements. By configuring evidence to combine the creation and end dating of evidence, ending an evidence record is no longer a two-step process that requires unnecessary clicks and extra navigation. If the end dating feature is enabled for an evidence type, an end dating option is displayed in the evidence creation wizard for that evidence type.

### Before you begin

For more information about the eligibility criteria that apply to the evidence end dating feature, see the *Reviewing and Processing Evidence Guide*.

### About this task

The following list outlines the scenarios in which an administrator can apply the end dating feature to evidence.

- **Dynamic evidence**

For dynamic evidence, configure the evidence end dating feature on the **New Dynamic Evidence Type** page and on the **Edit Dynamic Evidence** page.

- **Non-dynamic evidence**

For non-dynamic evidence, configure the evidence end dating feature on the **New Evidence Metadata** page and on the **Edit Evidence Metadata** page. Like other non-dynamic evidence pages, the appropriate wizard page must be generated by using the evidence generator, and then enabled by editing the server XML metadata file.

For more information, see the *Reviewing and Processing Evidence Guide* and the *Evidence Developers Guide*.

- **Custom evidence**

For more information about configuring the evidence end dating feature for new dynamic evidence types, see [Creating a dynamic evidence type version on page 15](#).

For more information about the technical implementation and behavior of the evidence wizard in relation to the end dating of evidence, see the *Evidence Developers Guide*.

The following procedure describes how to configure the end dating feature for dynamic evidence and for non-dynamic evidence.

### Procedure

1. Log on to Cúram as an administrative user.
2. Click **Administration Workspace**.
3. In the Shortcuts pane, click **Rules and Evidence**.
4. Depending on the evidence type that you want to configure the end dating feature for, click either **Dynamic Evidence** or **Non-Dynamic Evidence**.
5. For each evidence type, click **Edit** in the list actions menu.
6. Select **Enable End Dating when Creating Evidence**.
7. For non-dynamic evidence, when you select **Enable End Dating when Creating Evidence**, enter the page name for the wizard in **End Dating when Creating Evidence Wizard Page Name**.
8. Click **Save** to complete the changes.

## 1.2 Configuring dynamic evidence

By using the dynamic evidence editor in the Cúram administration application, administrators can define and maintain dynamic evidence types.

The process of creating and maintaining dynamic evidence types is an administrative rather than a development activity. Follow the proceeding example of how to create a dynamic evidence type, and then read further to familiarize yourself with more advanced dynamic evidence features. The proceeding outlines the different types of configurations that are available without redeploying the application or migrating data. By understanding the different types of configurations, it's possible to define the default configurations that are required for an implementation.

### Dynamic evidence types and dynamic evidence type versions

Dynamic evidence types are the fundamental elements of dynamic evidence. Dynamic evidence types are stored in the `EvidenceTypeDef` entity and link together all components that are involved in the administration of dynamic evidence.

Use dynamic evidence as an alternative to the traditional development-time definition and maintenance of evidence types, such as the non-dynamic evidence maintenance approach.

In the non-dynamic evidence maintenance approach, typically the following tasks are required before a new evidence type can be used in practice:

- Artefacts must be defined in the Cúram Development Environment.
- Java code must be written.
- Screens must be designed and specified.
- The entire application must be built, deployed, and tested.

In contrast, creating dynamic evidence types is strictly an administrative exercise. By using the Cúram administration suite and the dynamic evidence editor, administrators can dynamically define artefacts that are equivalent to the artefacts that a Cúram developer specifies of non-dynamic evidence. For more information about dynamic and static evidence, see [Evolution of dynamic evidence type metadata on page 25](#).

Dynamic evidence types are the equivalent of evidence types in non-dynamic evidence, that is, a grouping of related data elements to be recorded against a case. Dynamic evidence types interact with the Cúram evidence infrastructure in the same way that non-dynamic evidence types interact with the Cúram evidence infrastructure. Functionally, dynamic evidence types and non-dynamic evidence types largely equivalent.

In the Cúram Administration application, you can create dynamic evidence types and dynamic evidence type versions, and then configure the corresponding lifecycles and their association with case types.

Administration screens for dynamic evidence are in the Rules and Evidence section. To open a tab that shows a paginated list of all non-canceled dynamic evidence types that are on the system, click the Dynamic Evidence link in the Rules and Evidence section. Expand each of the dynamic evidence types to display a list of dynamic evidence type versions for the selected dynamic evidence type.

Dynamic evidence types are accessed and managed by using the **Dynamic Evidence List Page**, which is located in the Rules and Evidence section of the CúramAdministration application.

Dynamic evidence types can be linked to case types to indicate that information of this type is relevant and maintainable for a particular program. In essence, dynamic evidence types act as the container or the header record for all other elements and definitions necessary for dynamic evidence to function properly at run time. A dynamic evidence type must have one or more dynamic evidence type versions.

### Differences between dynamic evidence types and dynamic evidence type versions

Evidence is typically gathered in support of program rules, and such rules tend to evolve over time as policies are adjusted and refined. As a result, in many cases the evidence requirements for such programs can also evolve, as the requirement to collect more or less information about a case or person changes.

In the case of non-dynamic evidence, make such changes must be made in the Cúram Development Environment, tested and ultimately deployed into production. If evidence data structures in the newer version are changed, then some data migration might be required.

In contrast, dynamic evidence, supports this requirement for evidence definitions to evolve over time without the need for redeployment or data migration.

**Note:** However, the dynamic evidence infrastructure imposes restrictions on how metadata can change over time. For more information, see [Evolution of dynamic evidence type metadata on page 25](#).

For dynamic evidence definitions to evolve over time without the need for redeployment or data migration, dynamic evidence is designed with two main elements: dynamic evidence types and dynamic evidence type versions.

### ***Understanding dynamic evidence type versions***

A dynamic evidence type version applies from its effective date and is used to maintain evidence record data from this date onwards. However, unlike non-dynamic evidence multiple versions of dynamic evidence can exist in the system when all of these versions have different effective dates. There can only ever be a single dynamic evidence type version active for a particular effective date.

**Note:** If a caseworker attempts to enter evidence record for a dynamic evidence type for a date in the past, the system uses the dynamic evidence type version effective at that date. The date might be the latest version or it might be the version that was defined and in use previously. In this way, the system prompts only for information that is required at each point in the history of the evidence type. So, only relevant fields are displayed.

Dynamic evidence type versions store such information in the form of metadata, and in doing so provide the details necessary to generate evidence record pages for collecting and recording evidence data. Each data evidence record in respect of a dynamic evidence type is linked to the dynamic evidence type version used to create it.

Versions are recorded in the `EvidenceTypeVersionDef` entity where metadata is stored as a blob in XML format. Editing metadata is only supported through the dynamic evidence editor, which is provided with the dynamic evidence administration component. Modifying the XML metadata directly is not supported and can result in incorrect system behavior. Also, the structure of this XML metadata might change between releases of Cúram.

Dynamic evidence type versions are linked to dynamic evidence types in a many-to-one relationship. Each version belongs to one dynamic evidence type, and a dynamic evidence type can have many versions, each with different effective from dates.

Versions can be accessed by expanding individual dynamic evidence type list items on the **Dynamic Evidence Type List** page. The page displays a list of versions for the selected dynamic evidence types. The page is sorted by effective from date and is in descending order. If there is version of a dynamic evidence type with a status of `In Edit` status, the dynamic evidence type is placed at the start of the list. For more information, see [Dynamic evidence lifecycle on page 23](#). The following actions can be performed on dynamic evidence type versions:

- **View Metadata**

The action starts the dynamic evidence editor in a new tab to view the metadata for the selected version. Users can explore the metadata, but cannot save any changes, that is, the **Save** button and various other Palette buttons are unavailable.

- **Edit Metadata**

The action starts the dynamic evidence editor in a new tab to edit the metadata. In this mode, users can save the changes that users make. This action is available for dynamic evidence type versions only with a status of `In Edit`.

**Note:** A user, when using the editor, can access the same dynamic evidence type version record from two places: the dynamic evidence editor and the **Dynamic Evidence Type List** page. If the status of the edited dynamic evidence type version is changed through the administration page, for example by activating it, subsequent attempts to save metadata from the editor results in an error.

- **Edit Effective From**

Use the action to modify the effective from date of a dynamic evidence type version. The date is a mandatory property and cannot remain blank. The action is available for dynamic evidence type versions only with a status of `In Edit`.

- **New InEdit Copy**

The action copies the selected dynamic evidence type version and creates a new version of the dynamic evidence type version with a status of `In Edit`. The action is available for the latest dynamic evidence type version only with a status of `Active` in the list, and only where that dynamic evidence type has no versions already with a status of `In Edit`, that is, only the latest active dynamic evidence type version can be copied and extended. The structure of the new dynamic evidence type version metadata that uses the dynamic evidence editor can be changed, however, subject to the normal restrictions on the evolution of dynamic evidence type versions.

- **Activate**

Only dynamic evidence types with at least one `Active` version can be used as evidence record in a program. The **Activate** action marks a dynamic evidence type version as `Active`, and this action is available for all versions with a status of `In Edit`. When activated, a dynamic evidence type can be linked to a product or integrated case. As a result, an activated dynamic evidence type can be used in the caseworker workspace to record evidence record data.

**Note:** The activation process for dynamic evidence type versions uses Cúram deferred processing, which is asynchronous in nature. As a result, the user might be required to use the **Refresh** button on the **Dynamic Evidence Type List** page to see the relevant status turn from `PendingActivation` to `Active`.

When a dynamic evidence type version is activated, various extra artefacts are generated. For example, user interface tab configurations, CER Rulesets. For more information, see [Generated artefacts on page 104](#).

Activating a dynamic evidence type version is the last step before the version becomes live and evidence record data can be recorded in respect of the dynamic evidence type version. To ensure the validity of the version and metadata, a set of validations is automatically performed upon activation. Validation problems are displayed in the confirmation dialog that is displayed to the administrator on activation. All validation problems must be fixed before the dynamic evidence type version can be activated.

The four following validations are performed:

- A new dynamic evidence type version must have an effective date later than the effective date of the latest active version in the same dynamic evidence type, that is, the effective date cannot overlap with previous active versions.
- A new dynamic evidence type version must have an effective date that is later than the latest recorded evidence record for the previous active versions in the same dynamic evidence type, that is, the effective date cannot overlap with previously recorded evidence record data.

- The XML metadata is structurally validated against a predefined schema.
- Extra validations on the XML metadata that cannot be expressed in an XML schema, including cross version validations to enforce constraints on metadata evolution over time, are also performed.
- **Delete**  
The action deletes the selected dynamic evidence type version and all related artefacts that are generated upon activation. For example, dynamic UIM pages and resources that can be globalized, CER Rulesets, tab configurations. A dynamic evidence type version cannot be deleted if it is associated with **In Edit** or **Active** evidence records. Such evidence records must first be deleted to delete the dynamic evidence type version.

**Note:** The delete restriction is important when a user is testing new dynamic evidence type versions in a test or staging environment before a user puts the dynamic evidence type versions live in a production environment. When a user tests new versions, which are typically revisions to previous versions, all test evidence records for the previous versions must be deleted before their corresponding dynamic evidence type versions can be deleted.

### ***Creating a dynamic evidence type version***

A new dynamic evidence type version can be created by using the **New Version** action from the dynamic evidence type list item action menu. The **New Version** action is enabled only where the dynamic evidence type has no **Active** or **In Edit** dynamic evidence type versions.

When the **New Version** action from the dynamic evidence type list item action menu, a new modal window is displayed. In the modal window, the user must enter the effective from date for the new dynamic evidence type version. The new version is created with blank metadata.

### ***Creating a dynamic evidence type***

To create a dynamic evidence type, the user must click **New...** on the **Dynamic Evidence List** page. The **New Dynamic Evidence Type** page is displayed in a modal window.

A dynamic evidence type is the administrative equivalent to a non-dynamic evidence type. So, a dynamic evidence type is a logical grouping of related attributes about which an organization wants to record information in respect of a case. For example, income evidence, medical expense evidence, residency evidence.

**Note:** In non-dynamic evidence, evidence types are ultimately represented as individual database tables. In dynamic evidence, all dynamic evidence types are represented in the background by using a standard set of generic tables.

For example, an evidence type that is called Sample Income is to be defined. The evidence type consists of the following two attributes:

- Income Amount (a money value).
- Income Type (a drop-down list that contains different types of income).

To start to create this as a dynamic evidence type, perform the following actions:

1. Log in as an administrator.
2. Open the **Administration Workspace**.
3. In the shortcuts panel, select **Rules and Evidence**.



4. Click **Dynamic Evidence**. A list of all dynamic evidence types that are defined in the system are displayed.
5. Click **New...** and enter the following information:
  - **Evidence Type**: Sample Income
  - **Logical Name** SampleIncome
  - **Effective From**: 1/1/2011
  - **Security Group**: SAMPLE\_INCOME\_GROUP
6. Click **Save**.

The preceding steps create the overall structure for the Sample Income dynamic evidence type. More detail must then be added to this dynamic evidence type.

When the administrator creates a dynamic evidence type, the system automatically creates both a dynamic evidence type and a dynamic evidence type version in the background, saving the administrator time. The system automatically creates a dynamic evidence type and a dynamic evidence type version in the background because valid dynamic evidence types require at least one dynamic evidence type version.

## Page properties

The page has the six following properties:

- **Evidence Type**  
Evidence type is the display name of the dynamic evidence type. Unlike other properties, the name is not stored in the `EvidenceTypeDef` entity. Instead, the name is stored in the `EvidenceType` code table and can be globalized by using the standard globalization process for code tables. The evidence type code for this code table is generated by the dynamic evidence infrastructure. The evidence type name is mandatory and must be unique across all dynamic and non-dynamic evidence types. The system enforces a check for uniqueness by verifying that the supplied name is not in use by other evidence types. The check is applied for all supported locales.
- **Logical Name**  
The logical name is a unique identifier for the dynamic evidence type on the system. The logical name is used to reference the dynamic evidence type from various other configuration components. For example, rule classes and resources that can be globalized. The logical name uses a specific naming format. The logical name must start with a lowercase or uppercase English alphabetic character and can contain lowercase and uppercase English alphabetic characters, numeric characters, and underscores. The logical name is a non-modifiable property. When the logical name is entered, it cannot be changed on the **Modify** page.
- **Effective From Date**  
The effective from date is the effective date that is used for the initial dynamic evidence type version that is created from the page.
- **Security Group**  
The security group property is used to create a Curam security group for the dynamic evidence type. Dynamic evidence type security identifiers for evidence record maintenance operations, that are generated by the dynamic evidence infrastructure, are added to the group. The security group name is also used to construct the names of generated security identifiers. For more information, see [Security identifiers and security groups on page 105](#).



**Note:** Rather than separate groups for each dynamic evidence type version, there is a single group for the dynamic evidence type. In practice, if a caseworker has permission to access one dynamic evidence type version then the caseworker has permission to access all dynamic evidence type versions for a dynamic evidence type.

The security group name is a non-modifiable property and must be unique across dynamic evidence types.

- **Definition**

The definition field represents a short piece of descriptive text that explains the dynamic evidence type. The following three properties apply to the definition field:

- The definition field cannot be globalized.
- The definition field is optional.
- The definition field is used for annotation purposes only. For example, the definition is not referred to from the evidence record screens.

- **Description**

The description field is a text property that can be globalized. The description field is displayed on the **New Evidence Type** search page in the caseworker workspace, that is, the page where users can select an evidence type to create an evidence record. If no value is provided for this property, the system defaults to displaying the evidence type name in the Description column of this page. For more information about how the description field can be globalized by using the evidence type properties resource, see [Evidence type version properties resources on page 94](#).

Every evidence type has an associated properties resource that can be globalized. The naming convention for the properties resources is as follows:

1. “DynEvd\_EvidenceType\_”.
2. The logical name of the dynamic evidence type.
3. “.properties”.

An example is *DynEvd\_EvidenceType\_<logicalName>.properties*. The description text that is entered on the **Create Dynamic Evidence Type** page is inserted in *DynEvd\_EvidenceType\_<logicalName>.properties* for the default locale. The property key name consists of the evidence type logical name followed by “.description”. For example, *<logicalName>.description*.

## Generated artefacts

When a dynamic evidence type is created, the following artefacts are generated:

- A new evidence type code is generated and inserted into the EvidenceType code table by using the evidence type name that is entered on the **Create** page as a code table item description. The new code table entry is recorded against the server locale. For more information, see [EvidenceType codetable entries on page 104](#).

**Warning** The generated dynamic evidence type code table codes must not be manually modified or removed by using the System Administration application. If generated dynamic evidence type codes are manually modified, proper functioning of the Curam system cannot be guaranteed.

**Warning** Auto-generated code table entries are generated for the server locale, but are displayed for the locale of the currently logged in user. Administrator users who are configuring dynamic evidence must operate in the same locale as the server. In multi-locale deployments, configuring dynamic evidence in the same locale as the server prevents globalization issues that might occur before auto-generated code table items are converted for all supported locales.

- An initial dynamic evidence type version for the new dynamic evidence type is generated. The initial dynamic evidence type version has blank metadata and its effective from date is set to the **Effective From** attribute that is entered on the **Create Dynamic Evidence Type** page.
- Security group and security identifiers for security administration are also generated. For more information, see [Security identifiers and security groups on page 105](#). A security group is created by using the name that is specified on the **Create Dynamic Evidence Type** page. In addition, operation security identifiers are created and added to the new security group. If a security group with the name “EVIDENCEGROUP” is on the system, the new security identifiers are also added to the group. The group can be created for convenience by administrators. Administrators can use the group to set the security rights for all dynamic evidence types simultaneously by adding the group to the appropriate security roles.

**Warning** The dynamic evidence type security group property cannot be modified. The property cannot be changed by using the **Dynamic Evidence Type Modify** page. Do not physically change the name of a generated security group in the System Administration application. If names of generated security groups are manually modified, proper functioning of the Curam system cannot be guaranteed.

### ***Editing and modifying a dynamic evidence type***

Dynamic evidence type details can be modified by using the **Edit** action in the dynamic evidence type list item action menu.

Selecting the **Edit** action opens the Modify page in a modal dialog. The three following dynamic evidence type properties can be modified:

- Evidence type name.
- Definition.
- Description.

The preceding rules that apply to these properties for the Create process also apply to the Modify process.

### **Modifying a dynamic evidence type**

Dynamic evidence types, unlike non-dynamic evidence types, can have multiple versions that vary over time. With non-dynamic evidence, a single evidence type has a fixed set of attributes and relationships during its existence. The attributes and relationships can be changed only in tandem with a redevelopment and data migration exercise.

However, dynamic evidence allows for information that was recorded for an evidence type to evolve in response to the evolution of legislative and administrative evidence requirements. Each version of a dynamic evidence type is effective from a particular date, and the version remains effective until the next version.

As part of creating a dynamic evidence type, the system also creates a default dynamic evidence type version. To be useful, a dynamic evidence type must have at least one evidence type version that is associated with it. Attributes and other details for the evidence type are defined at the level of the evidence type version.

To define attributes and other such details, start the dynamic evidence editor. Start the dynamic evidence editor by performing the following two steps:

1. Expand the flipper on the newly created dynamic evidence type.
2. In the **Actions** button in the list item, click **Edit Metadata**.

The next steps are to define two key aspects of a dynamic evidence type: the model and the user interface.




### ***Defining the model***

The first key dimension to a dynamic evidence type version is its model, which describes its structure and behavioral characteristics.

The model defines the following five characteristics:

- The model's attributes and the attributes' data types.
- The model's relationships to other dynamic evidence types.
- The model's validations.
- How an evidence record of this type is to be summarized in descriptions.
- Various other functional aspects.

When you start the dynamic evidence editor, the **Model** tab is displayed by default. In the proceeding example, two attributes that were specified in the requirements are created: Income Amount and Income Type.

1. Click the  icon to add a data attribute. A data attribute is the dynamic equivalent of a database column on a non-dynamic evidence entity, that is, a single piece of information to record for an evidence type.
2. In the **Data Attribute Properties** pane, enter the following three pieces of information:
  - **Attribute Name:** incomeAmount
  - **Attribute Type:** Money
  - **Mandatory:** <checked>
3. Click the  icon to add another data attribute.
4. In the **Data Attribute Properties** pane, enter the following two pieces of information:
  - **Attribute Name:** incomeType
  - **Attribute Type:** Codetable
5. Expand the **Codetable Options** flipper and enter the following information: **Codetable:** IncomeTypeCode.
6. Click the  icon to save your work.

For more information about the other options available in the model section of the dynamic evidence editor, see [Defining a model for a client data type version on page 34](#). The next step is to define the user interface for this dynamic evidence type version.



## Defining the user interface

Ultimately, caseworkers interact with evidence types when caseworkers try to enter information in the evidence types for cases where the evidence is configured.

For non-dynamic evidence, developers provide the screen definitions for each evidence type as input to the build process. However, for dynamic evidence, administrators specify the screen definition information by using the dynamic evidence editor. Ultimately, the screen definition information is used to generate the screens to create, modify, and view evidence records at run time through the evidence record dashboard.

In the example of the `Sample Income` dynamic evidence type version, a simple user interface is defined that is to be used on all maintenance screens, that is, create, modify and view. The newly added data attributes are placed near each other on a cluster. The infrastructure is responsible for handling everything else that happens to display these screens at run time.

To define a simple user interface, perform the following steps:

1. Click the **User Interface** tab of the dynamic evidence editor.
2. Click the icon  to add a data attribute cluster.
3. In the **Attribute Cluster Properties** pane, opposite **Title** specify `Income Details`. Use the default settings for all the other properties.
4. Select `Income Amount`, the first of the data attribute fields in the list. Drag `Income Amount` over the central blue area of the newly added `Income Details` cluster and release the mouse button. `Income Amount` is a new field on the cluster. When you drop the field onto the cluster, the data attribute that you dragged disappears from the **Data Attribute Fields** pane because the same field cannot be dropped onto a screen twice.
5. Select the new field. Opposite the **Field Properties** pane, enter **Label:** `Income Amount`. Use the default settings for all the other properties.
6. Select `IncomeType`, the last remaining data attribute field in the list. Drag `IncomeType` over the central blue area of the newly added `Income Details` cluster and release the mouse button. `IncomeType` is a new field on the cluster. The field that you dragged is shown as a drop-down list because that attribute that you dropped is of type `CodeTable`. Similarly, all other attribute types are represented with the appropriate user interface controls.
7. Select the new field. Opposite the **Field Properties** pane, enter **Label:** `Income Type`. Use the default settings for all the other properties.
8. Click the icon  to save your work.
9. Close the **Editor** tab.

## Activating the dynamic evidence type version

A dynamic evidence type version with a status of `In Edit` means that the dynamic evidence type version is not yet approved for use in a product or case. The status indicates that an administrator is still working on its definition.

Another difference between dynamic evidence types and non-dynamic evidence types is that the latter are developed rather than configured. It is assumed that the latter are available for use in a case when non-dynamic evidence types are deployed on a system, that is, they have no `In Edit` status.

In the sample income dynamic evidence type that is to be available for use in a case, the dynamic evidence type must have at least one `Active` dynamic evidence type version.

To ensure that the dynamic evidence type has at least one **Active** dynamic evidence type version, perform the three following steps:

1. Ensure that the **Dynamic Evidence** tab is in the foreground. For more information about closing the **Dynamic Evidence**, see [Creating a dynamic evidence type on page 15](#).
2. From the **Action** menu at the end of the evidence type version row, select **Activate...**
3. On the Activate Dynamic Evidence Type Version dialog box that is displayed, select **Yes**.

The **Status** column of the dynamic evidence type version is now **Pending Activation**. Later, if you refresh the page the status of page is displayed as **Active**. The dynamic evidence type can now be used in a case. The next step is to link this dynamic evidence type with a case type.

### **Associating dynamic evidence types**

To be usable in the caseworker workspace to capture evidence record data, dynamic evidence types must be associated with Product Delivery or Integrated Cases.

To associate dynamic evidence types with Product Delivery or Integrated Cases, use the same administration interface that is used to associate non-dynamic evidence types to case types.

### **Associating the dynamic evidence type with a case type**

You might already have at least one case type that is registered on your system with which you could associate your new dynamic evidence type. The proceeding example assumes that you want to create a new case type from the beginning.

**Note:** Dynamic evidence types, not the finer-grained dynamic evidence type versions, are associated with case types. If a dynamic evidence type has more than one version, then all of the versions are effectively associated with the case type. So, in this respect, the evolution of a dynamic evidence type is linked to the evolution of any case types with which the dynamic evidence type is associated.

To create a new case type, in this example an integrated case type, perform the following 10 steps:

1. In the Administration Workspace Shortcuts pane, select **Case > Integrated Cases**. The **Integrated Cases** page is displayed that lists all defined integrated case types.
2. Click the **New...** button. The New Integrated Case Type dialog box is displayed.
3. In the **Integrated Case Type:** field, enter Sample Integrated Case. Use the default settings for all the other fields.
4. Click **Save**.
5. In the Type column of the **Integrated Cases** list page, click the Sample Integrated Case link. The **Sample Integrated Case** homepage is displayed.
6. Click the **Evidence Types** tab.
7. Click the **Add Evidence...** button. The Add Existing Evidence Type dialog box is displayed.
8. In the **Evidence Type:** field, enter Sample Income.
9. In the **Category:** field, enter Income. Use the default settings for all the other fields.
10. Click **Save**.

All the required administration is complete. The next step is to create an evidence record as a caseworker to test the new dynamic evidence type.

### ***Creating evidence records for this dynamic evidence type***

To create evidence records for a Sample Income dynamic evidence type, first create an integrated case for the new Sample Integrated Case integrated case type.

To create an integrated case for the new Sample Integrated Case integrated case type, perform the following five steps:

1. Log in to the caseworker application as a caseworker. For example, by using the caseworker user name.
2. On the **Home** tab, click the Search for a Person... link. Enter the search criteria for the person and click **Search**.
3. Click the link for the appropriate person who is listed in the search criteria results. The **Person** tab is displayed.
4. On the **Actions** menu for the **Person** tab, click the **New Case...** menu item.
5. On the **New Case** dialog box, select **Sample Integrated Case** for the **Type** field and click **Save**. The **Integrated Case** tab for the newly created case is displayed.

A case against which evidence records can be recorded is now available and sample income evidence can be entered.

**Note:** From a caseworker perspective, there is no difference between dynamic evidence types and non-dynamic evidence types. For a caseworker, both evidence types are evidence that a caseworker can enter. How a particular evidence type is defined does not affect a caseworker.

To enter sample income evidence, perform the following five steps:

1. On the **Integrated Case** tab, click the **Evidence** tab. By default, the Evidence Dashboard displays all evidence types for the recorded evidence record.
2. Expand the + button for the Sample Income evidence type. The New pane is displayed. The screen that was defined by using the dynamic evidence editor for the sample income dynamic evidence type is now displayed.

**Note:** The dynamic evidence infrastructure in the background added a **Received Date** field and populated the appropriate code tables with code table descriptions and shows mandatory fields with an asterisk. The **Received Date** field is common to all evidence types.

3. Enter a valid money amount in the **Income Amount** field.
4. Enter any selection in the **Income Type** list.
5. Click **Save**. The entered information is written to the database as an evidence record. You created and tested a new dynamic evidence type.

### ***Deleting an evidence type***

To delete a dynamic evidence type, use the **Delete** action in the dynamic evidence type list item action menu.

Deleting a dynamic evidence type removes all generated artefacts from the system, including code table entries for the evidence type code, security information, globalized text, and so on.

If the dynamic evidence type is linked to products or integrated cases, the links are also removed. A dynamic evidence type can be deleted only where it has no **Active** or **In Edit** dynamic evidence type versions.

### **Viewing dynamic evidence type details**

To view the details of a dynamic evidence type, expand the evidence type list item and select the **Details** tab.

The **Details** tab displays a read-only view of the evidence type details.

## **Dynamic evidence lifecycle**

Dynamic evidence, unlike non-dynamic evidence, has an associated lifecycle. Whereas non-dynamic evidence types are always in effect active, dynamic evidence can be in an number of other states. There are two aspects to the dynamic evidence lifecycle: the dynamic evidence type lifecycle, and the dynamic evidence type versions lifecycle.

The Dynamic Evidence Type lifecycle is simple and has two possible internal statuses: `Active` and `Canceled`. The system manages the statuses and the management is effectively hidden from the user. When a Dynamic Evidence Type is created, it is automatically assumes a status of `Active`. Only Dynamic Evidence Types with a status of `Active` are available for use as evidence record.

Deleting a Dynamic Evidence Type changes its state to `Canceled`, i.e. The Dynamic Evidence Type is not physically removed from the `EvidenceTypeDef` entity. However, all related generated artefacts *are* physically deleted from the system. Canceled Dynamic Evidence Types cannot be reactivated, and are filtered out by the administration and case worker workspaces.

Dynamic Evidence Type Versions have a different lifecycle which includes more states, and at all times the state is visible to users and clearly identifies what actions can be performed on each Dynamic Evidence Type Version:

- **InEdit**

When a Dynamic Evidence Type Version is created, its initial status is `In Edit`. The `In Edit` status indicates that the Version is not completely specified, that is, the Version can still be modified. For `In Edit` Versions, no evidence records can be created. `In Edit` Versions do not affect the runtime case worker application. At any time, a Dynamic Evidence Type can have zero or one `In Edit` Version only. An `In Edit` Version can be deleted or activated only.

- **PendingActivation**

Activating a Dynamic Evidence Type Version changes its state to `Pending Activation`. This is a temporary state to allow time for generation and publishing of Rulesets. At the end of this process the Dynamic Evidence Type Version status is automatically changed to `Active`.

After activating a Dynamic Evidence Type Version, users may or may not see the `Pending Activation` status, i.e. It is possible to immediately see the Dynamic Evidence Type Version status go to `Active`. This depends on the time taken to generate and publish CER Rulesets, and also the time to refresh the administration page after activation. If a status of `Pending Activation` is displayed, the administration page will not refresh automatically when the state is internally changed to `Active`. In this case, users will need to manually refresh the administration page to reflect the change of status; this can be done using the **Refresh** button on the Dynamic Evidence Type List Page.



**Note:** The Activation process for Dynamic Evidence Type Versions uses Cúram Deferred Processing, which is asynchronous in nature. In case of failure, this deferred process changes the status of the Dynamic Evidence Type Version from `Pending Activation` back to `In Edit`. In this case, the administrator will be shown a red exclamation mark on the Dynamic Evidence list page item which will link to a dialog box showing the reasons for the failure during Activation. Administrators should ensure that prior to activation of a new Dynamic Evidence Type Version, CER rules are validated and published to avoid activation errors.

- **Active**

The `Active` state indicates that a Dynamic Evidence Type version is now live, and that it can be used to maintain evidence records in the case worker workspace. As mentioned earlier, for a Dynamic Evidence Type to be available to be linked to a Case Type, it must itself have a status of `Active` and also have at least one Dynamic Evidence Type Version which has a status of `Active`.

Once in this status, Dynamic Evidence Type Versions can only transition out of this state by being deleted.

- **Canceled**

When a Dynamic Evidence Type Version is deleted in the Dynamic Evidence Type List Page, its status is changed to `Canceled`. This status marks the end of life of the Version. Like Dynamic Evidence Types, Versions are not physically deleted from the `EvidenceTypeVersionDef` entity. Canceled Versions can no longer be used in either in the administration or case worker workspace, nor can they be recovered.

### ***Dates in runtime evidence record maintenance***

In the maintenance of evidence records for dynamic evidence types, and for non-dynamic evidence types, the crucial dates are the received date and the effective date of change.

Caseworkers can use different dynamic evidence type versions in an intuitive manner based on the same business dates that are used with non-dynamic evidence types. Administrators must be aware of how the effective dates for the different versions of dynamic evidence types can affect the user interfaces that is presented to caseworkers.

### **Creating evidence records**

All evidence create pages, for dynamic and non-dynamic evidence types, have a received date field that defaults to the current date. The received date marks the start of the period for which the evidence record is active. The received date is a mandatory field.

For Non-Dynamic Evidence Types, case workers can (unless custom validations preclude it) enter any date they choose as the received date. Dynamic Evidence however imposes some limitations on this. One such limitation is that the Received Date may not be before the Effective Date of the earliest version of the Dynamic Evidence Type. In effect, an evidence record cannot be valid before the Dynamic Evidence Type itself is valid, which makes sense.

If several Versions of the Dynamic Evidence Type for which the evidence record is being created exist, how does the system know which one is being created, and hence which user interface to present to the user? The system will always initially present the user interface for creating the Version of the Dynamic Evidence Type that is `Active` on the current date.

If the user, while creating the evidence record, modifies the Received Date so that it falls in a time period where a different Dynamic Evidence Type Version is `Active`, then the system will redirect the user to the appropriate user interface for the correct Dynamic Evidence Type Version.



The data that the user has already entered will be pre-populated on the page to which the user is redirected.

Therefore, the Received Date field is effectively the key to determining the Dynamic Evidence Type Version in respect of which the evidence record is created. The caseworker user should not have to concern themselves with this - they simply consider the business meaning of the date and the system will present them with the appropriate user interface for creating the evidence record structure that applies at that point in time.

### **Modifying in-edit evidence records**

Modifying an evidence record with a status of `In-Edit` is an extension of the create process. The record is changed in-place, but no new member of the evidence record's succession set is created.

The Received Date is present on the modify page too, and if it is changed it may result in the user being redirected to another modify page to finish editing the record, as described for the Create page. This will happen when the user attempts to save the evidence record. In this way `InEdit` evidence records can be modified to belong to a different Dynamic Evidence Type Version (although in practice, this is an edge case which should happen very rarely).

The modify page header also has a field for another important business date - Effective Date of Change - however this does not become valid until the evidence record has been Activated, and attempting to change it will result in an error being displayed. This behaviour has been retained is identical to that in respect of Non-Dynamic Evidence Types..

### **Modifying active evidence records**

A user can modify an active evidence record in two ways: correcting incorrect data and recording a change in a participant's circumstances. This mechanism hinges on the Effective Date of Change field.

The Effective Date of Change field, in business terms, records the start of a time period for which a version of the evidence record is valid. For instance, if the piece of evidence is a record of a Participant's employment, what happens if the client gets promoted? It is a continuation of their previous employment, so it does not require an entirely new employment record. The record of the client's new job role succeeds the previous record, which remains valid up until the point when the client got their promotion. That point in time is captured by the Effective Date of Change.

If the Effective Date of Change is entered, then the system treats the modification of the Active evidence as a change in circumstances. A new evidence record is created as part of the same succession set, and its Active period commences at the Effective Date of Change. For Dynamic Evidence Types, If the Dynamic Evidence Type Version that applies for the Effective Date of Change is not the same as that which applied for the previous version of the evidence record then, as before, the user will be redirected to the appropriate user interface for the applicable Dynamic Evidence Type Version.

## **Evolution of dynamic evidence type metadata**

Dynamic evidence types can evolve over time. For example, a change in legislation might require that a new evidence attribute must now be recorded, starting from a specified date. Dynamic evidence supports this requirement by using dynamic evidence type versions to record modifications to metadata over time.

Metadata changes are made by copying the latest Active version (using the Dynamic Evidence Type Version **New InEdit Copy** action) to create a new Dynamic Evidence Type Version with a

status of `In Edit`. The effective date of the new Version can be set by the administrator to the date when the new change is required to take effect as specified by the legislation. Users may edit the new `In Edit` Version metadata to make the appropriate change to the Evidence structure.

**Note:** It is only possible to create a copy of the latest Active Version. Previous Versions cannot be copied. This mechanism is designed to support the natural evolution of metadata over time, and to implement additional restrictions on modifications to metadata elements between Versions

As metadata evolves, some metadata elements cannot be freely modified between Versions, and certain limitations have been put in place by the Dynamic Evidence infrastructure. These limitations are not enforced by the Dynamic Evidence Editor, but a set of validations is performed upon activation of subsequent Dynamic Evidence Type Versions. The first Dynamic Evidence Type Version for a Dynamic Evidence Type, while in an `In Edit` state, allows for the modification of all metadata elements.

Once activated, however, the following restrictions are applied to subsequent Dynamic Evidence Type Versions:

- **Parents**  
Once added in the initial Dynamic Evidence Type Version, Optional and/or Mandatory Parents cannot later be removed. New Mandatory and/or Optional Parents can be added in later Versions.
  - **Attributes**  
Once defined in the initial Active Dynamic Evidence Type Version, the following Attribute properties cannot be changed in later Versions:
    - Data Type
    - Volatile
    - The class of an Attribute e.g. If an Attribute is defined as a Data Attribute in the initial Version, it cannot be changed to be a Calculated Attribute in later Versions.
- Finally, note that Attributes defined in the initial Version can be deleted in later Versions and re-introduced in subsequent Versions, but they can only be re-introduced with the same attribute category, data type and volatility as defined in the initial Version.
- **Business Start and End Date**  
The Evidence Business Start and End Dates can be set or be blank in the initial Version, but these cannot be changed in later Versions.
  - **New Dynamic Evidence Type Versions**  
When a new Dynamic Evidence Type is created, the system automatically creates a new Version for it with blank metadata. Subsequently, the following restrictions apply:
    - While the Dynamic Evidence Type has an `In Edit` Version, no new `In Edit` Versions can be added to it.
    - If the Dynamic Evidence Type only has `Active` Versions, it is possible to create a new `In Edit` Version by copying the latest `Active` Version using its **New InEdit Copy** action. Metadata from the latest `Active` Version is copied into the new Version.
    - If all `Active` and `In Edit` Versions are deleted from the Dynamic Evidence Type it is possible to create a new `In Edit` Version with blank metadata using the Dynamic Evidence Type **New Version** action.

It should be noted that the restrictions to modify Attributes, Parents and Business Start and End Dates are only applied if there is at least one `Active` Version in the Dynamic Evidence Type. If

at any time the Dynamic Evidence Type is in an `In Edit Version` only (e.g. By deletion of all `Active Versions`), then all such restrictions are dropped and the `In Edit Version` is considered as an initial Version.

## Security administration

Dynamic evidence security administration is relatively straightforward, and, in general, follows the standard Cúram security administration process. Dynamic evidence supports operation-level security, but does not support field-level security.

Security Groups and operation Security Identifiers are generated when a Dynamic Evidence Type is created (see [Creating a dynamic evidence type on page 15](#)). The Security Group Name specified on the Dynamic Evidence Type Create page is used to create a new Security Group specific to the new Dynamic Evidence Type. The generated Dynamic Evidence Type Security Identifiers are added to this Security Group.

One technical point of note: unlike other Cúram operation Security Identifiers (which are generated for modeled facade operations), Dynamic Evidence operation Security Identifiers are slightly different. There are no facade operations specific to each Dynamic Evidence Type in Dynamic Evidence (as Dynamic Evidence Types are defined at administration time, not development time). As such, all evidence record maintenance operations in respect of Dynamic Evidence Types are funneled through a single generic facade where the operation-level security is managed.

Because there are no real facade operations, the operation Security Identifier names are generated based on information from the Dynamic Evidence Type definition (see [Security identifiers and security groups on page 105](#)). Three Security Identifiers are generated for each Dynamic Evidence Type: one each for Create, Modify and View operations. These Security Identifiers are added to the Security Group created in respect of the Dynamic Evidence Type. The Dynamic Evidence Type Security Group and Security Identifiers can be managed in the Cúram administration application. For more information, see the *Organization Administration Guide*. Security Groups can be added to User Roles to give access rights for the maintenance of individual Dynamic Evidence Types.

A special security group named “EVIDENCEGROUP” can also be used to administer security for Dynamic Evidence Types. If this security group exists, all generated Dynamic Evidence Type Security Identifiers will be added to it. This group is a placeholder for all Dynamic Evidence Security Identifiers. It is intended as a convenience group that can be used to grant access to all Dynamic Evidence operations at once (which is typically of great use in demo scenarios).

**warning** The names of the Security Groups and the names of generated Security Identifiers for Dynamic Evidence Types should never be manually modified in the System Administration application. This will result in undefined system behavior.

**Note:** The “EVIDENCEGROUP” Security Group is not added automatically by the system if it does not exist. Administrators can create this Group if they so wish. If any Dynamic Evidence Types were created before creating this Security Group, they will not be automatically added to it. However, users can manually add the other Dynamic Evidence Types to this group without risk.

## Dynamic evidence editor overview

The dynamic evidence editor has two tabs: **Model** and **User Interface**. In the editor, you must define the **Model** (attributes and relationships) before you define the **User Interface** (clusters and fields).

When you first open the dynamic evidence editor, the **Model** tab has focus. Both tabs are structured similarly as shown in the following figure:

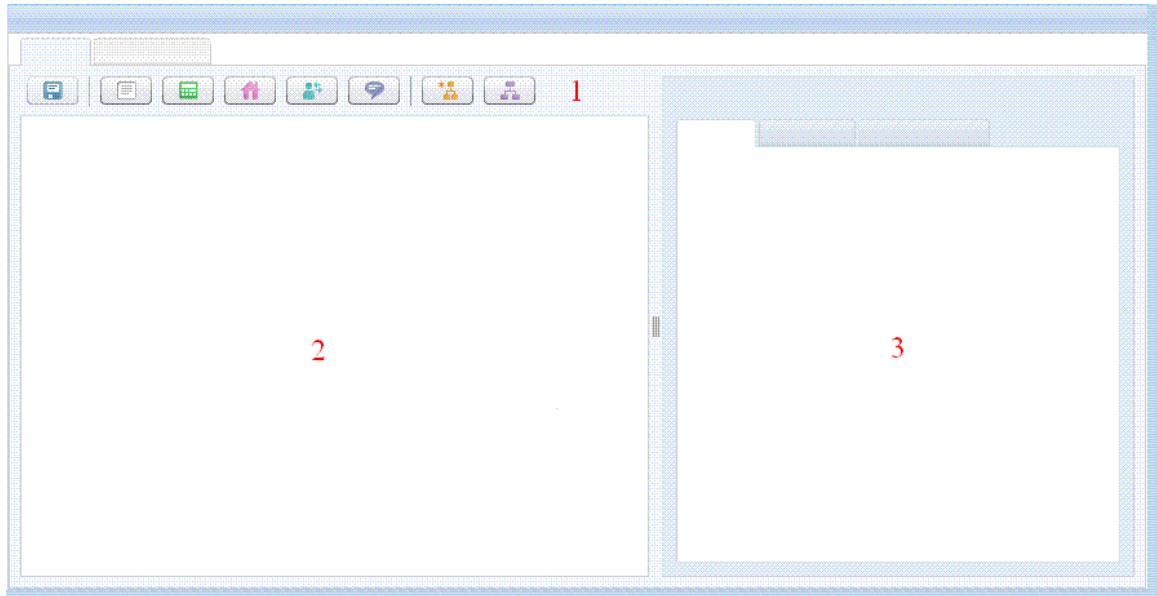


Figure 1: Dynamic Evidence Editor Structure

- **Palette (1) :**

The Palette is where you can create artefacts.

- For the **Model** tab, the Palette contains buttons to create Data Attributes, Calculated Attributes, Address Attributes, Related Case Participant Attributes, Related Employment Attributes, Comments Attribute, Optional Parents and Mandatory Parents.

- For the **User Interface** tab, the Palette is in the left frame of the Editor. The Palette contains a button to create a new Data Attribute Cluster and an accordion control that contains all of the attributes in the Model, categorized by type. You can drag attributes to the Canvas to create fields and clusters.

**Tip:** You cannot directly add a field to the user interface. Fields must relate to an existing attribute because the **Dynamic Evidence Editor** infrastructure uses the attribute definition to determine the associated screen widgets to display.

For both the **Model** and **User Interface** tabs, the Palette contains a **Save** button that you can use to save the current client data type version to the Cúram database.

- **Canvas (2)**

The Canvas shows the artefacts that you are modeling.

- For the **Model** tab, the Canvas graphically represents the client data type version that you are modeling, its underlying attributes, and parent and child relationships with other client data types.

The client data type version is represented as a shape or a class on the Canvas that you cannot remove. Attributes are represented in a list within this class, and relationships are represented as separate classes that are linked to the client data type version via lines.

- In the **User Interface** tab, the **Content** tab is to the right of the Palette and above the Properties Panel.

The Canvas is a view of how the Create, Modify and View screens for the client data type version might look in the user interface, and contains the clusters and fields that are currently defined.

**Note:** The user interface view is not an exact representation of runtime screens; it is intended for layout and planning purposes so that you can envisage what the Client Data screens might look like. For example, when generating screens for each client data type version, the client data infrastructure adds a range of additional fields and features such as Received Date for Create pages, Effective Date of Change and Change Reason for Modify Pages, Mandatory indicators for attributes marked as mandatory in the Model, and Save and Cancel buttons.

- **Properties Panel (3)**

The Properties Panel contains screens that you can use to add or change properties of a selected artefact. The properties that display in the Properties Panel depend on the type of artefact that you select. For example, the client data type version Properties Panel contains three more sub-panels of information.

- In the **Model** tab, the Properties panel is to the right of the Canvas. Properties screens exist for the client data type version, attribute types, and relationship types.
- For the **User Interface** tab, the Properties panel is under the Canvas. Properties screens exist for all cluster types and all field types.

## ***The dynamic evidence editor***

The dynamic evidence editor is a graphical editing environment for dynamic evidence type version metadata.

Administrators can use the dynamic evidence editor to define the storage, behavioral and visual characteristics of dynamic evidence type versions. These include:

- Model definitions and options for the Dynamic Evidence Type Version, including:
  - Attributes of the Dynamic Evidence Type Version, such as Data Attributes, Calculated Attributes, Address Attributes, Related Case Participant Attributes, Related Employment Attributes and Comments Attributes
  - Mandatory and Optional Parent Relationships between a Dynamic Evidence Type Version and other Dynamic Evidence Types
  - Standard Validations to be invoked at runtime when case workers enter data
  - Summary Details definitions which govern how an evidence record is described in the Evidence Workspace pages at runtime
  - Various behavioral characteristics of a Dynamic Evidence Type Version, such as the width of its create and modify pages at runtime, and Business Start and End dates for the Evidence Type
- User Interface definitions and options for the Dynamic Evidence Type Version, including
  - Clusters of Fields, such as Data Attribute Clusters, Address Clusters, Related Participant Clusters, Related Employment Clusters and Comments Clusters
  - Fields and Skip Fields

### ***Editor structure***

There is a logical order involved in defining the various components of dynamic evidence: it is always first necessary to define the model (attributes and relationships) before defining the user interface (clusters and fields).

This is because it is not possible to directly add a Field to the User Interface - the Field has to be in respect of an existing Attribute, as the Dynamic Evidence infrastructure will use the Attribute definition to work out what screen widgets to show in respect of each.

As such, the Dynamic Evidence Editor consists of two Tabs - Model and User Interface - and when first opened, the Model tab has focus.

Both tabs consist of three main structures (in slightly different configurations, but with the same basic meaning):

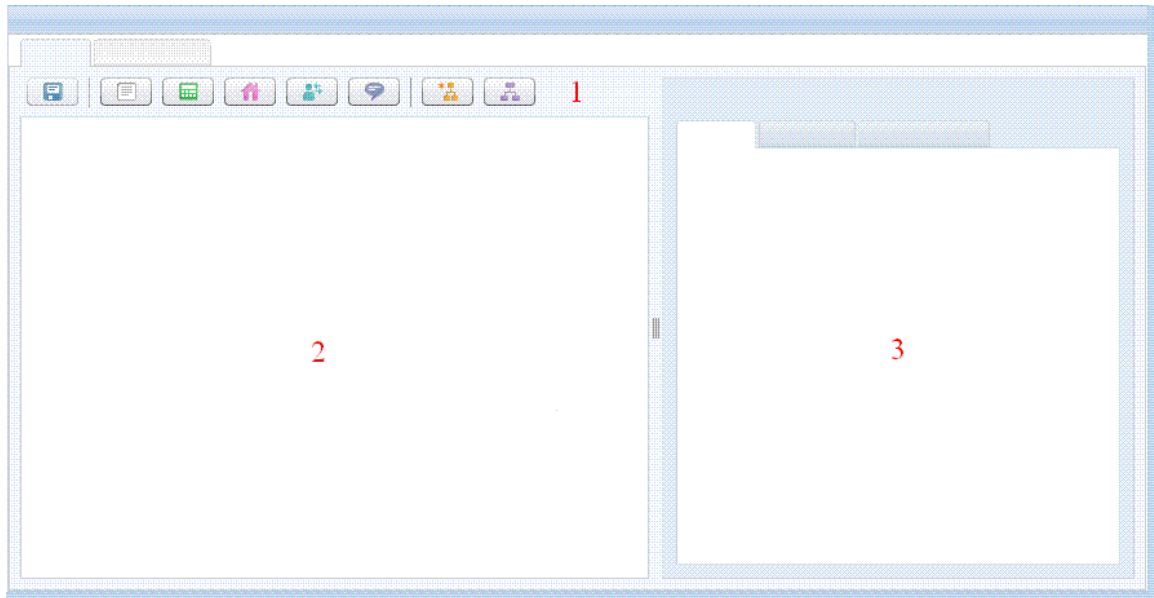


Figure 2: Dynamic Evidence Editor Structure

- **Palette (1) :**

The Palette is used to create artefacts.

The Palette for both the Model and User Interface Tabs each has a **Save** button, used to save the current Editor version of the Dynamic Evidence Type Version to the Cúram database.

- For the Model Tab, the Palette contains buttons to create Data Attributes, Calculated Attributes, Address Attributes, Related Case Participant Attributes, Related Employment Attributes, Comments Attribute, Optional Parents and Mandatory Parents.
- For the User Interface Tab, the Palette is more complex, having a button to create a new Data Attribute Cluster, but also containing an Accordion control containing all of the Attributes in the Model, categorized by type; these attributes can be dragged onto the Canvas to create Fields and Clusters. In the User Interface Tab, the Palette is located to left of the Editor.



- **Canvas (2)**

The Canvas contains a graphical representation of the artefacts being modeled.

- For the Model Tab, the Canvas contains a representation of the Dynamic Evidence Type Version being modeled, together with representations of all Attributes in respect of it, and all Parent/Child Relationships it has to other Dynamic Evidence Types.

The Dynamic Evidence Type Version is represented as a shape or a class on the Canvas which is always present - it is not possible to remove this. Attributes are represented in a list within this class, and Relationships are represented as separate classes linked to the Dynamic Evidence Type Version via lines.

- In the User Interface Tab, the Content tab is located to the right of the Palette, and above the Properties Panel.

Here, the Canvas contains a graphical representation of what the User Interface for the Create, Modify and View screens for this Dynamic Evidence Type Version are going to look like at evidence record maintenance time, containing all currently defined Clusters and Fields.

**Note:** The User Interface view is not an exact "what you see is what you get" (WYSIWYG) representation of runtime screens; rather it is used for layout and planning purposes, to get a feel for what the Evidence screens will look like. In particular, the Dynamic Evidence infrastructure, when generating screens for each Dynamic Evidence Type Version, will add in a range of additional fields and features such as Received Date for Create pages, Effective Date of Change and Change Reason for Modify Pages, Mandatory indicators for attributes marked as mandatory in the Model, Save and Cancel Buttons, etc.

- **Properties Panel (3)**

The Properties Panel is used to house screens that allow an administrator to add or change various properties of the currently selected artefact (see the next section for more information about selections). The Properties Panel will change its available properties dependent on the type of artefact currently selected (for example, the Dynamic Evidence Type Version Properties panel is relatively complex, having a further three sub-panels of information which can be maintained).

- For the Model Tab, Properties screens exist for the Dynamic Evidence Type Version, all Attribute types and all Relationship Types, and the Properties Panel is located to the right of the Canvas.
- For the User Interface Tab, Properties screens exist for all Cluster Types and all Field types, and the Properties Panel sits below the Canvas.

### ***Properties data display***

When you select an item in the dynamic evidence editor, your item selection governs the pre-populated data that displays in the **Properties Panel**.

You can select the following artefacts:

- All Model Attributes
- All Model Parents
- Client Data Type Version
- All Clusters



- All Data and Calculated Attribute Fields

You can select an artefact by clicking on it once. Typically, when you hover the mouse over an item without clicking it, the items that you can select are outlined in the Editor. When you select an item, the artefact reflects a different color.

**Tip:** An item that is you select is shared between the **Model** and **User Interface** tabs. Therefore, if you select an item in the **Model** tab and subsequently select an item in the **User Interface** tab, the original item is no longer selected when you return to the **Model** tab.

In some cases, you might select an artefact at the same time as you create a new artefact to immediately change its name or some other property.

Once you select an artefact, the Properties Panel updates to reflect the configurable properties for the artefact type with pre-populated values.

**Note:** You might select an artefact that does not have any configurable properties. For example, if you select attributes that can be dragged in the User Interface palette, the Properties panel contains no properties.

### ***Changing properties***

Generally, properties screens consist of a set of property names and property values. The values are typically maintained by an administrator. Where a value cannot be set by an administrator, the value is not even editable in the editor.

In order to change the value of a property, simply type in the new value and either hit the 'Enter' or 'Tab' keys. Hitting the 'Enter' key will in most cases update the value and keep the focus in the current property value field (the exception being multi-line fields like 'Description' properties where 'Enter' will add a carriage return to the property value), whereas hitting the 'Tab' key will update the value and move the focus on to the next property value field, if any.

**Note:** Switching the currently selected item to a different artefact while changing a field will attempt to commit the value before switching to the new Properties screen.

**Note:** Remember to hit the 'Save' button to persist your Editor work to the Cúram database!

### **Validation problems**

Every property value is validated when committed. Different properties have different associated naming and formatting rules, but in each case the administrator is informed if a proposed property value fails a validation.

In the case of validation failures:

- The property will not be updated to the new value. The property field will still contain the (invalid) value entered by the administrator to give them a chance to update it, but behind the scenes the Dynamic Evidence Type Version Metadata will not be updated.
- The property value field will be surrounded with a red border (to indicate invalid content) until the invalid value is cleared or updated to be a valid value.

- Putting the mouse over the red field results in the validation error message appearing in a red popup area. This message also contains information on the format to use to get the property value to pass validation.

### Defining a model for a client data type version

To define metadata for a client data type version, you must define the model in the **Dynamic Evidence Editor** as a first step. A model is the set of attributes and relationships that governs the structure of the evidence record that you want to store at run time.

In the **Model** tab of the **Dynamic Evidence Editor**, you can add a new artefact and add it to the client data type version class in the Canvas area of the Editor. You can model and maintain the following artefacts:

- Properties and information that is specific to the client data type version, in the Properties Panel
- Attributes of the client data type version and its relationships to other client data types, in the Palette area of the Editor

The following topics address how to configure properties, attributes, and relationships from your perspective as an administrator. The definitions that you specify in the Editor determines how the end user will interact with the software application.

#### *Client data type version properties*

In the **Properties Panel** of the dynamic evidence editor, you can configure the properties of a client data type version, define the property validations to be invoked at runtime, and define the summary details that display at runtime.

You can complete your configuration tasks on the following tabs in the **Properties Panel**:

- General Properties
- Validations
- Summary Details.

#### **General Properties**

To configure properties that determine how the end user interacts with the software application, use the **General Properties** tab in the Properties Panel of the dynamic evidence editor.

The following table lists the properties that you can configure.

Table 1: General Properties

Property name	Description
Read only	Configures a client data type to be read-only.  This property disables create and modify operations for an evidence record that is associated with a read-only evidence type. The worker can manually delete read-only evidence records.

Property name	Description
Correction Only	<p>Configures updates to an evidence record to allow only corrections.</p> <p>If you set this property to true, a case worker can correct the evidence record but cannot create a succession to an associated evidence record. In this case, The <b>Effective Date of Change</b> field is removed from the modify page for the evidence record.</p>
Save and New	<p>Adds a <b>Save and New</b> button to an evidence create page for the client data type.</p> <p>This property configures functions at runtime that allows the case worker to complete these tasks:</p> <ul style="list-style-type: none"> <li>• Save data that is currently entered to the database but not dismiss the page, and</li> <li>• Empty fields of data to allow data entry for another evidence record</li> </ul>
Related CP Attribute	<p>Optional: Sets the participant attribute on the Evidence Descriptor record to be the selected Related Case Participant at runtime.</p> <p>If you do not set this property, and the client data type has no Mandatory or Optional Parent relationships, the participant field on the Evidence Descriptor record is the Primary Client of the Case. If you do not set this property, and the Dynamic Evidence Type has at least one Mandatory or Optional Parent relationship, the client data infrastructure iterates through the parent hierarchy until a suitable participant is identified, for example, parent, grandparent, and so on.</p> <p>In order to avail of the Multiple Participant Evidence Update, this field must be set.</p>
Business Start Date	<p>Optional: Sets the business start date to be one of the data attributes that is defined for the current client data type version, with a data type of <code>Date</code>.</p> <p>When CER rules process data propagation, this property determines the start date from which to start propagating data for this client data type. For more information about rules data propagation, see <a href="#">Eligibility and entitlement rule sets on page 92</a>.</p>
Business End Date	<p>Optional: Sets the business end date to be one of the data attributes that is defined for the current client data type version, with a data type of <code>Date</code>.</p> <p>When CER rules process data propagation, this property determines the end date after which data should not be propagated for this client data type. For more information about rules data propagation, see <a href="#">Eligibility and entitlement rule sets on page 92</a>.</p>
Create Dialog Width	<p>Optional: Configures the display width, in pixels, of the Create page for the client data type version.</p> <p>If you do not set this property, the default value is 600 pixels.</p>
Modify Dialog Width	<p>Optional: Configures the display width, in pixels, of the Modify page for the client data type version.</p> <p>If you do not set this property, the default value is 600 pixels.</p>

Property name	Description
Calculated Attributes Rule Set Name	Optional: Sets the name of the CER Rule Set to be used in the runtime evaluation of Calculated Attribute values.  The Dynamic Evidence Editor treats this property as an optional property. However, if one or more Calculated Attributes are defined for the client data type version, you must set this property to be able to activate the client data type version. If you set the property, the property must be the name of a valid CER rule set.
Description	Optional: Sets a description for this client data type version that cannot be localized.  This property is for administrative annotation purposes only. The client data infrastructure does not use the property and the Evidence screens in the case worker workspace do not display it at runtime.
Online Help	Optional: Displays the online help text for the evidence create, evidence modify, and evidence view pages at run time. The online help text can be localized.

## Validations

Validations are data validity checks on evidence records that must be implemented before the records can be saved or activated. To configure validations, use the **Validations** tab in the Properties Panel of the dynamic evidence editor.

In Non-Dynamic Evidence, validations are implemented at development time by using Java code. In Dynamic Evidence, however, you must configure validations to occur at runtime assuming the client data type version has defined validations for the following scenarios:

- On saving data on an evidence create page
- On saving data on an evidence modify page
- On selecting **Apply Changes** or **Validate Changes** from the Evidence Workspace page

Separately or in combination, you can define the following client data validations for a client data type version:

- **Standard Validations**

By default, the **Dynamic Evidence Editor** provides a set of configurable Standard Validations. Based on an analysis of evidence validations over time, these Standard Validations are found to be the validations that are frequently used in evidence processing.

- **Additional Validations**

To specify a validation that cannot be expressed by using a Standard Validation, you can define a CER rule set to process validations. If you define additional validations, the CER validation rule set runs in addition to Standard Validations.

### *Standard validations*

The **Validations** tab of the Evidence Properties Panel, when opened, shows a list of all currently defined Standard Validations in respect of the current dynamic evidence type version.

In this list is a column containing the type of the Validation, plus a column which provides a narrative of the Validation that will be executed at runtime. Standard Validations are typically defined in terms of Model Attributes in the current Dynamic Evidence Type Version.

From the Validations tab, two buttons control the creation and deletion of Standard Validations:

- **Add**

Clicking this button brings up the **Add Validation** dialog. This dialog allows for the creation of a Standard Validation for this Dynamic Evidence Type Version. The functionality provided by this dialog is described in the following section.

- **Delete**

Only enabled when a Validation is selected in the Standard Validations list on the Validations tab, clicking this button removes the currently selected Validation from the Dynamic Evidence Type Version.

Also on the Validations Tab is a panel for Additional Validations. This panel allows the administrator to provide the name of a valid CER Validations Rule Set which will be executed at Case Evidence creation and modification time in addition to the list of Standard Validations. By default the radio button which indicates that there are no additional Validations defined for this Dynamic Evidence Type Version (labeled **None**) is selected; when this is selected, the Dynamic Evidence infrastructure will not look for a CER Validation Rule Set to execute.

Selecting the other radio button (**Use Rule Set**) enables the **Rule Set Name** field on the **Additional Validations** panel. This free form text field allows administrators to specify a CER Rule Set to be used as a Validation Rule Set for this Dynamic Evidence Type Version; for more information, see [Validation rule sets on page 87](#). If this field is filled in, it must contain the name of a valid CER Rule Set before this Dynamic Evidence Type Version can be activated.

**Note:** The options available for specifying Validations are not mutually exclusive; it is possible to define both Standard Validations and CER Rule Set Validations for a single Dynamic Evidence Type Version. With Summary Details, however, these options are mutually exclusive: it is only possible to select one of the preceding options.

### *Standard validation types*

Standard validations that you can configure include comparison validations, dependency validations, date of birth validations, and duplicate validations.

The following sections describe each validation type in detail.

### **Comparison Validation**

In a Comparison Validation, a data attribute is compared with another data attribute or a literal by comparing operators and attribute values.

Comparison Validation Property	Description
Source Field	The attribute whose value is to be compared. The Source Field can be a Data Attribute, Calculated Attribute or a Related Case Participant Attribute.
Comparison	The operator to use in the comparison; for more information, see the table that describes available operators.

Comparison Validation Property	Description
Target Field	<p>The attribute whose value is to be compared. As a rule, the Target Field must be of the same type (either Attribute Type or Attribute Data Type) to be comparable, and the Target Field list is filtered to only display attributes that are valid for comparison with the selected Source Field. One exception to this rule is for data attributes with a data type of Integer, Money or Float; these numeric types are mutually comparable.</p> <p>The Target Field must not point to the same attribute as the Source Field</p> <div> <b>Restriction:</b> You cannot use Related Employment Attributes, Address Attributes and Comments Attributes in Comparison Validations. </div>

The following table describes valid combinations of operators and data types for Data Attributes and Calculated Attributes. Because the behavior of Related Case Participant Attributes differs, the detail of those attributes are described in the next information table.

Table 2: Supported operators and applicable Data/Calculated Attribute data types in Comparison Validations

Operator	Applicable Data Types	Description
==	Boolean, String, Integer, Float, Money, Codetable and Date.	The <i>equal to</i> operator checks that Source and Target Fields have exactly the same value; if values differ, the validation check fails. More information follows about using this operator for Date fields.
<>	Boolean, String, Integer, Float, Money, Codetable, Date and Date Time.	The <i>not equal to</i> operator checks that Source and Destination Fields do not have exactly the same value; if values are the same, the validation check fails. More information follows about using this operator for Date fields.
<	Integer, Float, Money, Date and Date Time.	The <i>less than</i> operator checks that the Source Field value is less than the Destination Field value. If the Source Field value is greater than or equal to the Destination Field value, the validation check fails.
<=	Integer, Float, Money, Date and Date Time.	The <i>less than or equal to</i> operator checks that the Source Field value is less or equal to than the Destination Field value . If the Source Field value is greater than the Destination Field value, the validation check fails.

Operator	Applicable Data Types	Description
>	Integer, Float, Money, Date and Date Time	The <i>greater than</i> operator checks that the Source Field value is greater than the Destination Field value. If the Source Field value is less than or equal to the Destination Field value, the validation check fails.
>=	Integer, Float, Money, Date and Date Time	The <i>greater than or equal to</i> operator checks that the Source Field value is greater than or equal to the Destination Field value. If the Source Field value is less than the Destination Field value, the validation check fails.
before	Date and Date Time.	The <i>before</i> operator checks that the Source Field value is before the Destination Field value. If the Source Field value is on or after the Destination Field value, the validation check fails.
on or before	Date and Date Time.	The <i>on or before</i> operator checks that the Source Field value is on or before to the Destination Field value. If the Source Field value is after the Destination Field value, the validation check fails.
after	Date and Date Time	The <i>after</i> operator checks that the Source Field value is after the Destination Field value. If the Source Field value is on or before to the Destination Field value, the validation check fails.
on or after	Date and Date Time	The <i>on or after</i> operator checks that the Source Field value is after or equal to the Destination Field value. If the Source Field value is before the Destination Field value, the validation check fails.

**Note:** When the Source Field is populated with a data attribute that has a data type of `Date`, two additional attributes that do not exist in the client data type version metadata are added to the Target Field list:

- **evidenceReceivedDate**  
This attribute represents the date on which an agency receives a piece of Evidence into the organization. The Received Date is stored on the evidence descriptor at runtime and is frequently used in evidence comparison validations. The client data infrastructure automatically adds this field to every Create and Modify evidence page that relates to client data type version.
- **evidenceEffectiveDateOfChange**  
This attribute represents the effective date of change for the evidence record. The Effective Date of Change is stored on the evidence descriptor at runtime and is frequently used in evidence comparison validations. The client data infrastructure automatically adds this field to every Modify evidence page that relates to a client data type version.

The following table describes the operators that apply to Related Case Participant Attributes in Comparison Validations.

Table 3: Supported operators for Related Case Participant Attributes in Comparison Validations

Operator	Description
==	<p>The <i>Equal To</i> operator checks that Source and Target Fields represent the same participant; if field values do not equate to the same participant, the validation fails. An additional boolean attribute, <code>shallow</code> is provided for this validation but the client data infrastructure ignores it when the operator is <code>==</code>.</p> <p>If the Related Case Participant ID is the same, the underlying Concern Role ID must also be the same.</p>
<>	<p>The <i>Not Equal To</i> operator checks that Source and Destination Fields do not have exactly the same value; if field values are the same, the validation check fails. An additional boolean attribute called <code>shallow</code> is provided for this Validation. If <code>shallow</code> is checked in the <b>Create Validation</b> dialog, only the Related Case Participant IDs on the Evidence Record are compared. If <code>shallow</code> is not checked in the <b>Create Validation</b> dialog, the underlying Concern Role IDs are also checked for equality.</p>

The following table describes more options for Comparison Validation.



Table 4: Additional options for Comparison Validation

Options	Description
Literals	<p>A source attribute (Data Attributes or Calculated Attributes) can be compared against literals. As a rule, the literal value must be the same data type as the selected source attribute to be comparable. To compare a source attribute against a literal, select the <b>Use Literal</b> check box; subsequently you can type the literal value in the Target Fidel or select the value in the case of data types such as <code>Codetable</code> or <code>Boolean</code> or <code>Date</code>.</p> <div> <b>Tip:</b> You might need to select a code table item as a literal value when the data type of the source attributes is <code>Codetable</code>. </div> <p>If the data type of the first attribute or second attribute is <code>Boolean</code>, specify values of <code>true</code> or <code>false</code>.</p> <p>If the data type of the attribute is <code>Date</code>, specify the date value or select the date by using the Date Picker.</p> <p>For numeric data types such as Integer, Float and Money, you can enter locale-specific format for the literal value. For the Money attribute, you can also type the currency symbol.</p> <div> <b>Restriction:</b> Specify literal values only for data attributes. </div>
Multiple Clauses	<p>By selecting the <b>Multiple Clause</b> check box, you can specify multiple clauses in a Comparison Validation. Each clause must succeed for the overall validation check to succeed.</p> <p>To control the creation and deletion of Multiple Clauses, use these functions:</p> <ul style="list-style-type: none"> <li> <b>Add</b> <p>By clicking this button, you can add a clause to the current validation based on the source, target and operator fields that are selected.</p> </li> <li> <b>Delete</b> <p>Only enabled when a clause is selected in the Clause list, clicking this button removes the clause/comparison validation that is selected.</p> </li> </ul>
Message ID	<p>By clicking the <b>Add Validation Message</b> dialog adjacent to the Message property, you can set a custom validation message. More information follows about the Custom Validation Message for Comparison Validations.</p> <div> <b>Restriction:</b> If a validation has Multiple Clauses, this property is mandatory. </div>

The following table describes the mandatory properties for Multiple Clauses in Comparison Validations.

Table 5: Multiple Clause Properties

Multiple Clause Properties	Description
Conjunctions	<p>Controls whether any clause or all the clauses in a group are validated at run time.</p> <ul style="list-style-type: none"> <li>If you select the <b>Any Clause</b> radio button, and if any one of the clauses passes at runtime, the entire validation check succeeds.</li> <li>If you select the <b>All Clauses</b> radio button, all clauses must succeed for the entire validation check to succeed.</li> </ul>

### Dependency Validation

Use the Dependency Validation to enforce a dependency of a particular type between two attributes.

**Restriction:** You cannot use Calculated Attributes in Dependency Validations.

Dependency Validation Property	Description
First Attribute	The Data Attribute, Address Attribute, Related Case Participant Attribute or Comments Attribute on which the Second Attribute depends.
Second Attribute	The Data Attribute, Address Attribute, Related Case Participant Attribute or Comments Attribute that depends on the First Attribute.

Dependency Validation Property	Description
Dependency	<p>The nature of the dependency that can include one of the following values:</p> <ul style="list-style-type: none"> <li> <b>Must enter second attribute</b>            If you select this value, and subsequently the case worker enters field value for the attribute pointed to by the First Attribute, they must also enter a field value for the attribute pointed to by the Second Attribute. If the case worker enters a value in the first field but not in the second field, this validation check fails.         </li> <li> <b>Must not enter second attribute</b>            If you select this value, and subsequently the case worker enters a field value for the attribute pointed to by the First Attribute, they must not enter a field value for the attribute pointed to by the Second Attribute. If the case worker enters values in both fields, this validation check fails.         </li> <li> <b>At least one attribute</b>            If you select this value, subsequently the case worker must enter a value into either or both fields pointed to by First Attribute and Second Attribute. If the case worker leaves both fields empty, the validation check fails.         </li> <li> <b>Only one attribute</b>            If you select this value, subsequently the case worker must enter a value into one or other fields pointed to by First Attribute and Second Attribute. If the case worker enters values in both fields or does not enter values in either field, this validation check fails.         </li> </ul>
Bidirectional	<p>This boolean property applies to Dependency Validations with a defined dependency of <b>Must enter second attribute</b> and <b>Must not enter second attribute</b> only. Selecting the Bidirectional property causes the words <b>...And Vice Versa</b> to be added to the descriptions under <b>Dependency</b>.</p> <div data-bbox="869 1372 1437 1464" style="border: 1px solid blue; padding: 5px;"> <p><b>Restriction:</b> You cannot use this property when the other dependency values are selected.</p> </div>

The following table describes more options in Dependency Validation.

Table 6: Additional options in Dependency Validation

Options	Description
Literals	<p>You can specify literal values for both the First Attribute and the Second Attribute. As a rule, the literal value must be the same data type as the selected first attribute or second attribute. To specify literals either to an First Attribute or Second Attribute, select the <b>Use Literal</b> checkbox; subsequently you can type the literal value in the Source or Target Literal Field A source attribute (Data Attributes or Calculated Attributes) can be compared against literals. As a rule, the literal value must be the same data type as the selected source attribute to be comparable.</p> <div><b>Tip:</b> You might need to select a code table item as a literal value when the data type of the source attributes is <code>Codetable</code>.</div> <p>If the data type of the first attribute or second attribute is <code>Boolean</code>, specify values of <code>true</code> or <code>false</code>.</p> <p>If the data type of the attribute is <code>Date</code>, specify the date value or select the date by using the Date Picker.</p> <p>For numeric data types such as Integer, Float and Money, you can enter locale-specific format for the literal value. For the Money attribute, you can also type the currency symbol.</p> <div><b>Restriction:</b> Specify literal values only for data attributes.</div>
Message ID	<p>By clicking the <b>Add Validation Message</b> dialog adjacent to the Message property, you can set a custom validation message. More information follows about the Custom Validation Message for Comparison Validations.</p>

Date of Birth Validation

In Date of Birth Validations, the date of birth of the Participant that is pointed to by a Related Case Participant Attribute in the client data type version is verified to be on or before a specific date.

**Restriction:** You cannot use Calculated Attributes in Date of Birth Validations.

Date of Birth Validation Property	Description
Related Participant	<p>The Related Case Participant whose date of birth value is to be used in the comparison.</p> <p>The choicelist for this property is pre-populated with all Related Case Participant Attributes that are currently defined for the client data type version. At runtime, the date of birth for the Person pointed to by the Related Case Participant is compared against the date that is specified in the <b>Input Date</b> field. If the <b>Input Date</b> is before the date of birth, the validation check fails.</p> <div> <b>Restriction:</b> Only Related Case Participants of type <code>Person</code> are valid for use in Date of Birth validations, even though the <b>Dynamic Evidence Editor</b> does not enforce this restriction. </div>
Input Date	<p>Data attributes with a data type of <code>Date</code> to be used in the comparison. Two additional attributes that do not exist in the client data type version metadata are added to the Input Date field list:</p> <ul style="list-style-type: none"> <li> <b>evidenceReceivedDate</b> <p>This attribute represents the date on which an agency receives a piece of Evidence into the organization. The Received Date is stored on the evidence descriptor at runtime and is frequently used in evidence comparison validations. The client data infrastructure automatically adds this field to every Create and Modify evidence page that relates to client data type version.</p> </li> <li> <b>evidenceEffectiveDateOfChange</b> <p>This attribute represents the effective date of change for an evidence record. The Effective Date of Change is stored on the evidence descriptor at runtime and is frequently used in evidence comparison validations. The client data infrastructure automatically adds this field to every Modify evidence page that relates to a client data type version.</p> </li> </ul>

The following table describes more options in Date of Birth Validation.

Table 7: Additional options in Date of Birth Validation

Options	Description
Message ID	<p>By clicking the <b>Add Validation Message</b> dialog adjacent to the Message property, you can set a custom validation message. More information follows about the Custom Validation Message for Comparison Validations.</p>

## Duplicate Validation

In Duplicate Validations, evidence records that are verified, according to specified criteria, as duplicates are prevented from being recorded on the system.

To identify duplicates, the Duplication Validation can check different sets of evidence records. For example, if the client data type version has one or more parent client data types, the Duplicate Validation only checks child records of parent records at runtime, that is, the sibling records of the current record..

However, if the client data type version has no parent relationships, the Duplicate Validation checks all evidence records for duplicates at run time.

**Restriction:** You cannot use Calculated Attributes in Duplicate Validations.

Duplicate Validation Property	Description
Use Date Range, Start Date, End Date	<p>If you select the <b>Use Date Range</b> check box, two mandatory properties display on the <b>Create Validation</b> dialog: Start Date and End Date. These properties must point to a data attribute with data type of <code>Date</code>.</p> <p>At runtime, the Duplicate Validation identifies any records in the selection set that have attribute values pointed to by the Start and End Dates, and are equal to the values that are specified in the Create or Modify evidence screens. If duplicate records are returned, the validation check fails.</p>
Other Attributes to Check	<p>Optional: a list of other attributes whose values are to be compared to check for duplicates (in tandem with any Date Range provided)</p> <p>If records in the selection set have attribute values that are equal to the values for the attributes in the <b>Other Attributes to Check</b> list (specified in the Create or Modify evidence screens), the validation check fails.</p> <div><p><b>Note:</b> If multiple attributes are in the list, can take the following actions:</p><ul style="list-style-type: none"><li>To check the attributes individually, select <b>Check each attributes individually</b>.</li><li>To check attributes in combination, that is, by checking that all the attributes in the selection list are unique, select <b>Check attributes together</b>.</li></ul></div>

Duplicate Validation Property	Description
Validate Date Range and Other Attributes together	<p>If you select this property, at runtime, the Duplicate Validation checks for evidence records that have overlapping Start and End dates and attribute values that are equal to the values for attributes in the <b>Other Attributes to Check</b> list (specified in the Create or Modify evidence screens). If duplicate evidence records are returned, the validation check fails.</p> <p>If you do not select this property, at run-time, the Duplicate Validation either checks for evidence records that have overlapping Start and End dates or attribute values that are equal to the values for attributes in the <b>Other Attributes to Check</b> list (specified in the Create or Modify evidence screens). If duplicate evidence records are returned, the validation check fails.</p>

The following table describes some additional available options in Duplicate Validation.

Table 8: Additional options in Duplicate Validation

Options	Description
Message ID	By clicking the <b>Add Validation Message</b> dialog adjacent to the Message property, you can set a custom validation message for both Date Range attributes and for other attributes. More information follows about the Custom Validation Message for Comparison Validations.

## Custom Validation Message

To set a custom validation message for an validation type, specify the following properties:

Table 9: Custom validation message properties

Validation Message Mapping Properties	Description
Message	<p>The text to use for the validation message.</p> <p>In the <b>Messages Parameters</b> property list, you can specify parameters with attribute names in this format: opening curly brace, parameter number, closing curly brace, for example {0}, {1}, and so on . If a validation check fails at runtime, the parameters that you specify are substituted in the validation message that displays.</p> <p>For example, suppose that the Message property is set as follows: {0} must not be equal to true and {1} must not be equal to SX2. The Message Parameters are set as follows: isPregnant, gender.</p> <p>If the comparison validation fails at run time, after a user creates an evidence record, the message that displays to the user is similar to the following: isPregnant must be equal to true and gender must be equal to female.</p>
Message ID	<p>A mandatory string to be used as the key to the Message property value. The string can be any valid identifier, for example, MyEvidenceTypeVersion.ComparisonValidation.Message</p>

Validation Message Mapping Properties	Description
Message Parameters	<p>An ordered list of Data Attributes, Related Case Participant Attributes, Address Attributes, Comments Attributes, or Calculated Attributes to be used in the Message.</p> <p>For Comparison Validations or Date of Birth Validations, two additional attributes that do not exist in the client data type version metadata are added to the field choicelists: <code>evidenceReceivedDate</code> and <code>evidenceEffectiveDateOfChange</code>. See the preceding sections for more details about these attributes.</p>

### Summary Details

The Summary Details (also known as Summary Information) for a dynamic evidence type version specifies the value to be used in the Description column of the Dynamic Evidence Type Workspace page for each evidence record in respect of this dynamic evidence type version.

The Dynamic Evidence Type Workspace page is the tab opened by clicking on an Evidence Type name on the Evidence Dashboard. The Description appears on the Dynamic Evidence Type Workspace page as a hyperlink that, when clicked, opens up the Business Object Tab for the selected evidence record.

Dynamic Evidence provides a number of different ways to specify Summary Details for a Dynamic Evidence Type Version, controlled by the radio buttons on the Summary Details Tab of the Evidence Properties Panel.

The following options are provided:

- **None**

No Summary Details have been provided for the Dynamic Evidence Type Version. At runtime, the Description for evidence records in respect of this Dynamic Evidence Type Version will be represented as an asterisk (to allow the hyperlink to be clicked). However, it is strongly recommended that Summary Details be provided for every Dynamic Evidence Type Version, as it would otherwise be impossible for case workers to differentiate evidence records for such Dynamic Evidence Type Versions.

- **Use Rule Set**

As with Validations, it is possible to use a CER Rule Set to calculate the Summary Details description at runtime, and this is typically used in the case where Summary Mappings cannot be used to achieve the desired outcome.

If this option is selected, the **Rule Set Name** field gets enabled for editing. As the name suggests, the Rule Set Name should be populated with the name of a valid CER Summary Rule Set if the Summary Details for the Dynamic Evidence Type Version is to be calculated using CER Rules. Activating the Dynamic Evidence Type Version will validate that any value provided for this property points to a valid CER Rule Set.

Information on defining Summary Details using CER can be found in the chapter on CER Rule Sets; see [Summary information rule sets on page 85](#) for more details.

- **Use Mapping**

The most usual (and straightforward) way of defining Summary Details for a Dynamic Evidence Type Version is by defining Summary Mapping information. Using Summary Mapping allows the administrator to specify the description to be used in the case worker workspace as a Resource Message, and also allows the administrator to specify a number of



attributes to be used in the display of the Dynamic Evidence Type Workspace pages. Summary Mapping is described in detail in the next section.

### Summary Mapping

If **Summary Mapping** is selected as the means of providing Summary Details for the dynamic evidence type version, administrators can set optional properties.

Table 10: Summary Mapping Properties

Summary Mapping Property	Description
Start Date	<p>If this property is set in the Editor, the value of the Attribute pointed to by the Start Date property will be used in the Dynamic Evidence Type Workspace page as the first date in the <b>Period</b> column of the list for each evidence record. If this property is not set in the Editor, the system will attempt to use the value for the Evidence Business Start Date property as the first date in the <b>Period</b> column. If, in turn, this date is also not set, then the Case Start Date will be used as the first date in the <b>Period</b> column at runtime.</p> <p>In the Editor, this property can be set to any Data Attribute or Calculated Attribute with a data type of <i>Date</i>.</p>
End Date	<p>If this property is set in the Editor, the value of the Attribute pointed to by the End Date property will be used in the Dynamic Evidence Type Workspace page as the second date in the <b>Period</b> column of the list for each evidence record. If this property is not set in the Editor, the system will attempt to use the value of the Evidence Business Start Date property as the second date in the <b>Period</b> column. If, in turn, this date is also not set, the Period will be displayed as open-ended i.e. Without a second date displayed in the <b>Period</b> column of the list for each evidence record (e.g. 2/2/2001 -)</p> <p>In the Editor, this property can be set to any Data Attribute or Calculated Attribute with a data type of <i>Date</i>.</p>
Participant	<p>If this property is set in the Editor, the name of the participant pointed to by the Participant property will be used to populate the Participant column of the Dynamic Evidence Type Workspace page at runtime. If this property is not set in the Editor, the name of the Primary Client will be populated instead.</p>
Message ID	<p>To set the text to be used for the Description column of the Dynamic Evidence Type Workspace page, administrators must set the <i>Message ID</i> property. To set this property, the search icon adjacent to the <i>Message ID</i> property should be clicked; this opens the <b>Edit Summary Message</b> dialog.</p>

### Edit Summary Message Dialog

To set or modify the Message for a Summary Mapping, the following properties must be specified:

Table 11: Summary Mapping Properties

Summary Mapping Property	Description
Message	<p>The text to use for the Summary Details Message. This Message can be parameterized with Attribute values, placeholders for which are specified in the Message with the format of: opening curly brace, parameter number, closing curly brace - e.g. {0}.</p> <p>At runtime, the parameters specified in the following <code>Message Parameters</code> list will be substituted into the Message to construct the Evidence Description to display. See the <code>Message Parameters</code> property for more information on parameterization of messages.</p>
Message ID	<p>Mandatory string to be used as the key to the Message property value; can be any valid identifier (e.g. <code>MyEvidenceTypeVersion.SummaryMessage</code>)</p> <p>In the Editor, this property can be set to any Data Attribute or Calculated Attribute with a data type of <code>Date</code>.</p>
Message Parameters	<p>An ordered list of Data Attribute and/or Calculated Attributes or Address Attribute to be used in the Message. Placeholders are put in the Message to indicate that an Attribute value should be substituted into the Message at runtime, and these placeholders follow the following pattern: {0}, {1}, {2}, etc.</p> <p>For example, say the <code>Message</code> property is set to the following:</p> <p>Earns \${0} {1} working for {2}</p> <p>And the <code>Message Parameters</code> are set the following:</p> <p>incomeAmount, incomeFrequency, employerName</p> <p>Then the message (given the appropriate data in the evidence record) could be displayed as something like:</p> <p>Earns \$123.50 weekly working for Midway Services Inc.</p>

**Note:** For each supported locale, you must also implement currency symbol, placement, and spacing rules in the relevant properties files at development time.

For each language that is installed in the system, you must add the currency symbol to the localized text in the relevant properties file. For example, if English and French are installed, you can add the currency symbol, placement, and spacing rules as follows.

1. For example, add the following entry to the English properties file:







```
IncomeSummary.Text=${0} per month
```

2. Add the following corresponding entry to the French properties file:

```
IncomeSummary.Text={0} $ par mois
```

## Adding new attributes

Use the **Model Palette** to add evidence attributes of various types to a dynamic evidence type version.

-  Add Data Attribute
-  Add Calculated Attributes
-  Add Address Attribute
-  Add Related Case Participant Attribute
-  Add Related Employment Attribute
-  Add Comments Attribute

The following sections describe each of these in detail.

### New Data Attribute

Use **New Data Attribute** to create a new data attribute for the dynamic evidence type version. The new attribute is displayed in the Model Canvas, where you can select it, and open the Properties Panel for Data Attributes.

This panel allows the administrator to set the following properties:

Table 12: Data Attribute Properties

Property name	Description
Attribute Name	<p>This mandatory property sets the Model name for the selected Data Attribute. The value of this property is treated as the internal identifier for the Data Attribute across the entire Dynamic Evidence Type Version meta data, and as such must be unique across all Attributes in the Dynamic Evidence Type Version. The Attribute Name is never used in the generation of screens for evidence record pages, and so the case worker will never see its value.</p> <p>Attribute Names must follow a specific naming format:</p> <ul style="list-style-type: none"> <li>• They must start with a lowercase English alphabetic character</li> <li>• They can only contain lowercase or uppercase English alphabetic characters, numeric characters and underscores.</li> <li>• They must not contain any reserved words such 'relatedEmployment' or 'comments' as these are reserved identifier in the Dynamic Evidence Editor.</li> </ul>

Property name	Description
Attribute Type	<p>This property sets the data type for the selected Data Attribute.</p> <p>The following Data Types are available:</p> <ul style="list-style-type: none"> <li>• String (default) - any character value</li> <li>• Integer - whole numbers</li> <li>• Date - a calendar date</li> <li>• DateTime - a calendar date and time</li> <li>• Money - a currency value</li> <li>• Boolean - true or false</li> <li>• Float - a floating-point number</li> <li>• Codetable - a Cúramcodetable code</li> </ul> <p>The Attribute Type fundamentally affects many aspects of the Data Attributes administrative and runtime behavior:</p> <ul style="list-style-type: none"> <li>• It governs what Type Safety Validations will get executed for values entered into a Field in respect of this Attribute; all evidence record attribute values are validated by the Dynamic Evidence infrastructure before they can be saved to the database to ensure that they are valid values for the specified Type.</li> <li>• It governs what User Interface widget is used to display the Attribute Value, both at administration time (in the User Interface tab of the Editor) and at runtime (on evidence record create, modify and view screens in respect of this Dynamic Evidence Type Version).</li> <li>• It controls the Attributes available for selection in Standard Validations, Summary Details, General Evidence Properties, etc.</li> <li>• It determines the data type options available in the relevant Attribute Properties panels</li> <li>• And many others - where the behavior of Dynamic Evidence is affected by the Attribute Type, this is documented separately in this guide</li> </ul> <p>Attempting to change the Attribute Type of a Data Attribute which is already referenced somewhere in the Dynamic Evidence Type Version (e.g. It is on the User Interface, or referenced in Validations, or referenced in Summary Information, etc.) will cause the Editor to confirm with the administrator whether or not they want to proceed. If they choose to proceed, the Editor will remove all references to it (e.g. In Business Start and End Dates, Validations, Summary Information, etc.), and will update the User Interface based on the new Type.</p>

Property name	Description
Domain Name	<p>Domain Name can be created with custom widget implementation and it should be added to the Curam Domain list. For more details on developing Domain and its custom widget, see the <i>Custom Widget Development Guide</i>.</p> <div> <p><b>Note:</b></p> <ul style="list-style-type: none"> <li>The domain (custom widget) should be configured with edit-renderer, view-renderer and converter</li> <li>The domain converter (custom widget) should be derived from the Standard Domain like <code>SvrStringConverter</code>, <code>SvrInt32Converter</code>, etc.</li> <li>The target value on the edit-renderer (widget implementation) should be in basic data type format</li> <li>Custom widget with multiple hierarchies is not supported</li> </ul> </div>
Default Value	<p>This optional property sets the default value to be pre-populated on evidence record create pages for the selected Data Attribute. Default values must be valid for the selected Attribute Type, and the Editor will not allow default values with incorrect formats for the selected Attribute Type to be set by the administrator.</p> <div> <p><b>Note:</b> In most cases, the Default Value is a free from text field, but this changed for the following attributes.</p> <p>drop down containing 'true' and 'false' where the Attribute Type is set to 'Boolean'.</p> <p>drop down containing code table items where the Attribute Type is set to 'Codetable'.</p> <p>text field with date picker where the Attribute Type is set to 'Date'.</p> <p>Locale specific format can be typed in for the data types such as Date, Integer, Float and Money and typing locale specific Currency Symbol can be typed in case of Money attribute.</p> </div>

Property name	Description
Volatile	<p>This property setting indicates whether values for this Data Attribute can logically change over time, and is currently solely used in CER Ruleset processing; this property has no effect on the runtime evidence record screens.</p> <p>For example, a 'Vehicle' Evidence Type may have two attributes - vehiclePurchasePrice and vehicleFairMarketValue. In this example, vehiclePurchasePrice is non-volatile (it has a value at evidence record creation time, and this value cannot ever change), whereas vehicleFairMarketValue is volatile (the fair market value will likely decrease over time).</p> <p>If set to True, the Volatile Property will cause this Attribute to be generated as a Time line Attribute in generated Dynamic Evidence Data Rule Sets, allowing its value to change across succession sets. If set to False, the Attribute will be generated as a non-time line Attribute which can have a single value over time.</p> <p>The Volatile property cannot be changed between Dynamic Evidence Type Versions for a Dynamic Evidence Type; once this property is set in a Dynamic Evidence Type Version and that Version is activated, it must continue to have the same value for all subsequent Dynamic Evidence Type Versions.</p>
Mandatory	<p>This property setting indicates that the selected Attribute should be considered Mandatory in evidence record create and modify pages. This property does not apply to Boolean or Codetable Data Attributes, Calculated Attributes or Related Case Participant Attributes. If set, the caseworker must specify a value for Fields in respect of this Data Attribute, and such Fields will be displayed with an asterisk to indicate to the case worker that they are required. Note that all Mandatory Attribute must appear in the User Interface!</p>
Calculate on create If Blank	<p>This property is used to specify a calculated attribute to be used to populate this data attribute if it is blank on create. A drop-down box enables linking the Data Attribute to a Calculated Attribute. The drop-down lists Calculated Attributes in the Evidence Type that are of the same data type as the Data Attribute. Once a Calculated Attribute is selected it will be used to calculate the value of the Data Attribute if it is blank on the Create page.</p>
<<Data Type>> Options	<p>Data Attributes have a number of options that can be applied to them that govern their behavior. These vary for each Attribute Type. The options are described in <a href="#">Data Attribute and Calculated Attribute type options on page 59</a>.</p>
Description	<p>This property configures a non-localizable Model description value for the selected Attribute. This is for annotative purposes only, and is never displayed to the case worker at runtime.</p>

### Add Calculated Attribute

Calculated attributes are dynamic evidence type version attributes. Rather than being provided by the caseworker, calculated attributes are calculated by the system at runtime and are read-only. Values for calculated attributes are evaluated using the execution of a CER Calculated Attributes Rule Set.

Clicking on the **Add Calculated Attribute** button in the Model Palette creates a new Calculated Attribute for the Dynamic Evidence Type Version, displays it in the Model Canvas, selects it, and opens the Properties Panel for Calculated Attributes.

All Attribute property settings relevant to Data Attributes are also relevant to Calculated Attributes with the exception of Default Value, Volatile and Mandatory settings (in general, any properties which affect the editing of the Attribute do not apply to Calculated Attributes as they are read-only in nature). Please refer to the Data Attributes property settings for more details.

As mentioned in the section on Evidence Properties, the Rule Set Name for Calculated Attributes is configured on the General tab of the Dynamic Evidence Type Properties Panel. For more information on the use of CER Rule Sets for Calculated Attributes, please refer to [Calculated attributes rule sets on page 91](#).

### Add Address Attribute

An address attribute is a single field that represents a complete address that can be maintained by caseworkers at runtime. Address attributes can be dragged as user interface clusters, where the attributes expand into a complete set of address fields, for example, address line 1, country, and so on.

Clicking on the **Add Address Attribute** button in the Model Palette creates a new Address Attribute for the Dynamic Evidence Type Version, displays it in the Model Canvas, selects it, and opens the Properties Panel for Address Attributes. This panel contains the following properties:

Table 13: Address Attribute Properties

Property name	Description
Attribute Name	<p>This property sets the Model name for the selected Address Attribute. The value of this property is treated as the internal identifier for the Address Attribute across the entire Dynamic Evidence Type Version meta data, and as such must be unique across all Attributes in the Dynamic Evidence Type Version. The Attribute Name is never used in the generation of screens for evidence record pages, and so the case worker will never see its value.</p> <p>Attribute Names must follow a specific naming format:</p> <ul style="list-style-type: none"> <li>• They must start with a lowercase English alphabetic character</li> <li>• They can only contain lowercase or uppercase English alphabetic characters, numeric characters and underscores.</li> <li>• They must not contain any reserved words such as <code>relatedEmployment</code> or <code>comments</code> as these are reserved identifier in the Dynamic Evidence Editor.</li> </ul>

Property name	Description
Volatile	<p>This property setting indicates whether values for this Address Attribute can logically change over time, and is currently solely used in CER Ruleset processing; this property has no effect on the runtime evidence record screens.</p> <p>For example, a <code>Vehicle</code> Evidence Type may have two attributes - <code>vehiclePurchasePrice</code> and <code>vehicleFairMarketValue</code>. In this example, <code>vehiclePurchasePrice</code> is non-volatile (it has a value at evidence record creation time, and this value cannot ever change), whereas <code>vehicleFairMarketValue</code> is volatile (the fair market value will likely decrease over time).</p> <p>If set to True, the Volatile Property will cause this Attribute to be generated as a Time line Attribute in generated Dynamic Evidence Data Rule Sets, allowing its value to change across succession sets. If set to False, the Attribute will be generated as a non-time line Attribute which can have a single value over time.</p> <p>The Volatile property cannot be changed between Dynamic Evidence Type Versions for a Dynamic Evidence Type; once this property is set in a Dynamic Evidence Type Version and that Version is activated, it must continue to have the same value for all subsequent Dynamic Evidence Type Versions.</p>
Mandatory	<p>This property setting indicates if the selected Attribute is Mandatory in evidence record create and modify pages. If set, the caseworker must enter information in the Address Field when the casework creates or modifies evidence records for this Dynamic Evidence Type Version.</p>
Description	<p>This property configures a non-localizable Model description value for the selected Attribute. This is for annotative purposes only, and is never displayed to the case worker at runtime.</p>

### Add related case participant

Related case participants are participants other than the Primary Client to be associated with an evidence record.

Related case participants allow case workers to do one of three things:

- Select an existing Case Participant (i.e. a Participant who has already been added to the Case) to associate with the current evidence record
- Search for a Participant (i.e. a Participant who has not already been added to the Case) to associate with the current evidence record (and also with the Case, creating a new Case Participant record at the same time)
- Register a new Case Participant of type Representative and associate them with the current evidence record, creating a new Case Participant record at the same time

In many cases it is sufficient that the evidence record be associated with an existing Case Participant, and so it is possible for the administrator to configure Related Case Participants so that only the first of these options is available to case workers.

Clicking on the **Add Related Case Participant Attribute** button in the Model Palette creates a new Related Case Participant Attribute for the Dynamic Evidence Type Version, displays it in the Model Canvas, selects it, and opens the Properties Panel for Related Case Participant Attributes.

This panel contains the following properties:



Table 14: Related Case Participant Attribute Properties

Property name	Description
Attribute Name	<p>This property sets the Model name for the selected Related Case Participant Attribute. The value of this property is treated as the internal identifier for the Related Case Participant Attribute across the entire Dynamic Evidence Type Version metadata, and as such must be unique across all Attributes in the Dynamic Evidence Type Version. The Attribute Name is never used in the generation of screens for evidence record pages, and so the case worker will never see its value.</p> <p>Attribute Names must follow a specific naming format:</p> <ul style="list-style-type: none"> <li>• They must start with a lowercase English alphabetic character</li> <li>• They can only contain lowercase or uppercase English alphabetic characters, numeric characters and underscores.</li> <li>• They must not contain any reserved words such <code>relatedEmployment</code> or <code>comments</code> as these are reserved identifier in the Dynamic Evidence Editor.</li> </ul>
Participant Type	<p>The case worker may want to register a new participant on the case for a Related Case Participant attribute value rather than selecting or searching for one that is currently registered on the System. When a new related case participant is required, basic participant details are entered and the System will create a Representative for this Participant. The Participant Type property is used to indicate the type of participant a Representative will represent. The setting of this property will not result in the creation of a Participant of that specified. For Example, A Person is not created if Participant Type = Person, however, a Representative is created to represent that person. The Participant Type setting specifies the type of participant that this representative will represent. The default value for representative <code>Type = CONTACT</code> for all participant type settings when creating the Representative. For more information about Representative types, see the <i>Participant Guide</i>.</p> <p>The following values can be selected for Participant Type:</p> <ul style="list-style-type: none"> <li>• Person</li> <li>• Employer</li> <li>• Service Provider</li> <li>• Unknown</li> </ul>
Participant Role Type	<p>This property, possible values for which are taken from the <code>CaseParticipantRoleType</code> codetable, specifies what type of Case Participant Role record will be created when the Related Case Participant Attribute is entered in the evidence record create or modify screens as runtime.</p>
Participant Type Options	<p>This property allows administrators to display only a subset of Case Participant Role Types in the second panel of the Related Case Participant Cluster (i.e. The panel which allows case workers to search for a Participant to associate with this evidence record). Possible values are taken from the <code>CaseParticipantRoleType</code> codetable.</p>

### Add related employment attribute

Related employment attributes represent employment records to be associated with evidence records.

Related Employment Attributes operate similarly to Mandatory and Optional Parent Relationships, in that they result in caseworkers being presented with a wizard for Evidence records creation. At run time, case workers will be presented with a list of employments from which they will select one to relate to the Evidence record; the final page in the wizard will then show the create page for the Evidence Type. On View and Modify Pages, the Related Employment Attribute will detail the employer and case participant for the related employment record.

Clicking on the **Add Related Employment** button in the Model Palette creates a new Related Employment Attribute for the current Dynamic Evidence Type Version, displays it in the Model Canvas, selects it and opens the Properties Panel for Related Employment Attribute. Only one Related Employment Attribute can exist in respect of a Dynamic Evidence Type Version.

Table 15: Related Employment Attribute Properties

Property name	Description
Attribute Name	For Related Employment Attributes, the Attribute name is always defaulted to <code>relatedEmployment</code> , which is a reserved identifier in the Dynamic Evidence Editor. This Attribute Name cannot be changed.
Description	This property configures a non-localizable Model description value for a selected Related Employment Attribute. This is for annotative purpose only, and is never displayed to the Case worker at run time.
Participant Type Options	At run time, Related Employment Attributes result in a <b>Select Employment</b> page in the evidence record creation wizard for case workers. This page shows a list of employment records, one of which has to be selected by the case worker before progressing to create an Evidence Record for the associated Evidence Type Version. By default, this list of employment records will consist of those for both the Primary Client and any Member Participant Role Types for Case Participants on the related case. It is however possible to refine this list by specifying a list of Case Participant Role Types (from the Case Participant Role Type code table) to be used to retrieve Employment records. Note that if any item is added to this list, the above defaults do not apply - i.e. it will be up to the administrator to manually add Primary Client and Member if this is desired.

### Add Comments Attribute

Click **Add Comments Attribute** to add a comments attribute to the dynamic evidence type version.

At runtime, for Comments Clusters in respect of such attributes, the case worker will see a multi-line free text comments field on the evidence record create and modify screens. In situations where this Attribute is being viewed, the text will be displayed as a label. Only one Comments Attribute can exist in respect of a Dynamic Evidence Type Version.

The following properties can be administered for Comments Attributes:

Table 16: Comments Attribute Properties

Property name	Description
Attribute Name	For Comments Attributes, the Attribute name is always defaulted to <code>comments</code> , which is a reserved identifier in the Dynamic Evidence Editor. This Attribute Name cannot be changed.
Description	This property configures a non-localizable Model description value for the selected Comments Attribute. This is for annotative purposes only, and is never displayed to the case worker at runtime.

### Data Attribute and Calculated Attribute type options

For data attributes and calculated attributes, the Attribute Type controls the additional options that are available for that attribute.

Changing the Attribute Type for a Data Attribute or Calculated Attribute will remove any previous options which had been set in respect of that Attribute, and cause the relevant new Options Property Panel to be displayed.

The following sections describe the Options properties available for each Attribute Data Type.

The Attribute Type fundamentally affects many aspects of the Data Attributes administrative and runtime behavior:

- It governs what Type Safety Validations will get executed for values entered into a Field in respect of this Attribute; all evidence record attribute values are validated by the Dynamic Evidence infrastructure before they can be saved to the database to ensure that they are valid values for the specified Type.
- It governs what User Interface widget is used to display the Attribute Value, both at administration time (in the User Interface tab of the Editor) and at runtime (on evidence record create, modify and view screens in respect of this Dynamic Evidence Type Version).
- It controls the Attributes available for selection in Standard Validations, Summary Details, General Evidence Properties, etc.
- It determines the data type options available in the relevant Attribute Properties panels
- And many others - where the behavior of Dynamic Evidence is affected by the Attribute Type, this is documented separately in this guide

### String

For data attributes with a data type of String, you can set specific properties.

Table 17: String Type Properties

Property name	Description
Minimum Length	This property sets the minimum character length for the Attribute when entered by the case worker. Entering a value with a lesser number of characters than the value of this property will result in a validation error at evidence record save time. Locale specific format can be typed in for Minimum Length.

Property name	Description
Maximum Length	This property sets the maximum character length for the Attribute when entered by the caseworker. Note that at runtime, case workers will not be able to physically enter more characters into a Field in respect of this Attribute than is specified in this property value. Locale specific format can be typed in for Maximum Length.
Compress Embedded Spaces	This property indicates that any extra whitespace embedded in the Attribute string, and that all leading and trailing whitespace, be removed from the value entered by the case worker before storing it. For more more information, see the <i>Server Developer's Guide</i> .
Remove Leading Spaces	This property indicates that any leading spaces should be stripped off the Attribute value entered by the case worker.
Remove Trailing Spaces	This property indicates that any trailing spaces should be stripped from the Attribute value entered by the case worker before storing it.
Convert to Uppercase	This property indicates that the contents of this Attribute value entered by the case worker should be converted to uppercase before storage.

### *Integer, Float, or Money*

For data attributes with a data type of `Integer`, `Float`, or `Money`, you can set specific properties.

Table 18: Numeric Type Properties

Property name	Description
Maximum Value	This property sets the maximum value for the Integer, Float or Money Attribute entered by the case worker. Locale specific format can be typed in for Maximum Value ( Currency Symbol can also be typed along with the Maximum Value in case of Money attribute).
Minimum Value	This property sets the minimum value for the Integer, Float or Money Attribute entered by the case worker. Locale specific format can be typed in for Minimum Value ( Currency Symbol can also be typed along with the Maximum Value in case of Money attribute).

### *Date*

For data attributes with a data type of `Date`, you can set specific properties.

Table 19: Date Type Properties

Property name	Description
Current Date Comparison	This property setting will perform a validation against the the value specified by the case worker for this Date, to ensure that it is before,after, equal to, not equal to, on or before, on or after to the current date.

Property name	Description
Custom Message	To set a custom validation message, the administrator must set the <code>Custom Message</code> property. To set this property, select the radio button <b>Current Date Comparison</b> and the search icon adjacent to the <code>Custom Message</code> property should be clicked: this brings up the <b>Add Validation Message</b> dialog. For more details on Custom Validation Message, see the Custom Validation Message section.

### *DateTime*

For data attributes with a data type of `DateTime`, you can set specific properties.

Table 20: *DateTime Type Properties*

Property name	Description
Current Date Comparison	This property setting will perform a validation against the the value specified by the case worker for this Date, to ensure that it is before, after, on or before, on or after to the current date time (note that it is not possible or sensible to check a <code>DateTime</code> for equality against the Current Date, and so <code>equal to</code> has been removed from the list of available operators).
Custom Message	To set a custom validation message, the administrator must set the <code>Custom Message</code> property. To set this property, select the radio button <b>Current Date Comparison</b> and the search icon adjacent to the <code>Custom Message</code> property should be clicked: this brings up the <b>Add Validation Message</b> dialog. For more details on Custom Validation Message, see the Custom Validation Message section.

### *Codetable*

For data attributes or calculated attributes with a data type of `Codetable`, you can set specific properties.

Table 21: *Codetable Type Properties*

Property name	Description
Codetable Name	This property sets the Codetable name for the Data Attribute or Calculated Attribute. At runtime, the case worker will be able to select a value for Fields in respect of this Attribute from a dropdown list of Codetable items for the Codetable whose name is specified in this property value.

### **Adding relationships**

Relationships for a dynamic evidence type version define an association between the version and other dynamic evidence types.



**Note:** Mandatory parents and optional parents' relationships are always made between a dynamic evidence type version and a dynamic evidence type. Structuring the relationships in this way means that relationships for a dynamic evidence type can evolve over time. For example, you can add a relationship in subsequent dynamic evidence type versions.

When you create an evidence record for a dynamic evidence type version with one or more parent relationships, you must select a parent record before the child record can be created.

In the example of two dynamic evidence types `Income` and `IncomeAllocation`, the dynamic evidence type version for `IncomeAllocation` is a mandatory parent relationship with `Income`. In evidence terms, `IncomeAllocation` is a child evidence type of `Income`. When a caseworker tries to create an evidence record of type `IncomeAllocation`, a wizard is displayed to the caseworker. The first page of the wizard contains a list of potential parent records of type `Income`. The caseworker must select a parent record of type `Income` to create the `IncomeAllocation` an evidence record.

**Note:** There are two supported relationship types in dynamic evidence: mandatory parents and optional parents. In relation to dynamic evidence type parent relationships, for either mandatory parents or optional parents the caseworker must select a parent record at run time. Optionality refers to the option of which parent dynamic evidence types to associate with the evidence record. Mandatory is a specific or enforced parent dynamic evidence type to associate with the evidence record. A dynamic evidence type that is defined as a child must have, at a minimum, one parent evidence specified.

Parent relationships are added to the dynamic evidence type version model definition that uses the following two buttons from the Model Palette:

-  Add Mandatory Parent
-  Add Optional Parent

### Add mandatory parent

Use **New Mandatory Parent Relationship** to add a new mandatory parent relationship to a dynamic evidence type version model definition.

After you create a new mandatory parent relationship to a dynamic evidence type version model definition, you create a shape in the model canvas that represents the new relationship, select the relationship, and open the Mandatory Parent Properties Panel. The dynamic evidence type version then, in essence, becomes a child evidence type of the parent evidence type.

The **New Mandatory Parent Relationship** button adds a new Mandatory Parent Relationship to a Dynamic Evidence Type Version Model definition, creates a shape in the Model Canvas representing the new Relationship, selects the Relationship, and opens up the Mandatory Parent Properties Panel. The Dynamic Evidence Type Version then, in essence, becomes a Child Evidence Type of the Parent Evidence Type.

Dynamic Evidence Type Versions can have multiple Mandatory Parent Relationships. Create pages for Evidence Types that define Mandatory Parents will be wizard based. Each Mandatory Parent will result in a new wizard page containing a selection list for that parent on the evidence record create page. The case workers will have to select one record as the parent record from each wizard page list (i.e. It is mandatory to select a parent record for each Mandatory Parent type).

It is not valid to have multiple Mandatory Parent Relationships to the same Dynamic Evidence Type for the same Dynamic Evidence Type Version, and the Evidence Editor will enforce this restriction.

The following properties can be set in respect of a Mandatory Parent Relationship:

Table 22: Mandatory Parent Properties

Property name	Description
Relationship Name	<p>This property sets the Model Relationship Name for the selected Mandatory Parent Relationship. The value of this property is treated as the internal identifier for the relationship and must be unique across all Relationships in the Dynamic Evidence Type Version. The Relationship Name is never used in the generation of screens for evidence record pages, and so the worker will never see its value. They are however used in the generation of CER Data and Processing Rule Sets for this Dynamic Evidence Type Version - see <a href="#">Eligibility and entitlement rule sets on page 92</a> for more details.</p> <p>Relationship Names must follow a specific naming format:</p> <ul style="list-style-type: none"> <li>• They must start with a lowercase English alphabetic character</li> <li>• They can only contain lowercase or uppercase English alphabetic characters, numeric characters and underscores</li> <li>• They must not contain any reserved words such as <code>relatedEmployment</code> or <code>comments</code> as these are reserved identifier in the Dynamic Evidence Editor.</li> </ul>
Parent Type Code	<p>This property setting specifies the Evidence Type Code for the Mandatory Parent. At runtime, the case worker must select one instance of an evidence record for the Parent Type Code during the creation of the child evidence record.</p>
Filter Attribute Name	<p>The attributes of Evidence Type Code for Mandatory Parent whose attribute value is to be filtered. The Field can be a Data Attribute of type String, Boolean, Integer, Money, Float and Date.</p>
Filter Attribute Literal Value	<p>A Filter Attribute can be compared against literals. As a general rule, the literal value must be of the same data type as that of the selected attribute in order to be comparable.</p> <p>To filter an attribute against a literal, select the filter attribute name from Filter Attribute Details; the literal value can then be typed (or selected in case of data types such as Codetable, Boolean or Date) in to the Filter Attribute Value field.</p> <div style="border: 1px solid blue; padding: 10px; margin-top: 10px;"> <p><b>Note:</b></p> <p>Literal Value can only be specified for Data Attributes.</p> <p>Administrators may need to select a code table item as a literal value when the data type of the selected filter attribute is "codetable".</p> <p>Where the data type of filter attribute is <code>Boolean</code>, the values of <code>true</code> and <code>false</code> can be provided.</p> <p>In case where the data type is Date, date value can either be typed or can be selected through a Date Picker. Locale specific format can be typed for the literal value for the numeric data types such as Integer, Float and Money attribute.</p> </div>
Description	<p>This property configures a non-localizable Model description value for the selected Mandatory Parent Relationship. This is for annotative purposes only, and is never displayed to the case worker at runtime.</p>



**Add optional parent**

Use **New Optional Parent Relationship** to add a new optional parent relationship to a dynamic evidence type version model definition.

After you create a new optional parent relationship to a dynamic evidence type version model definition, you create a class in the model canvas that represents the new relationship, select the relationship, and open the Optional Parent Properties Panel.

Dynamic Evidence Type Versions can have multiple Optional Parent Relationships. Create pages for Evidence Types that define Optional Parents will be wizard based. Evidence records for all of these Dynamic Evidence Types will be listed in a single list on a single wizard page that is part of the evidence record wizard for this Dynamic Evidence Type Version, and case workers will have to select one record from the list as the parent record. So, even though it is 'mandatory' to select one record in the evidence record maintenance pages, caseworkers have an option in which the evidence type caseworkers select as the evidence type of the parent record.

It is not valid to have multiple Optional Parent Relationships of the same Evidence Type for the same Dynamic Evidence Type Version, and the Evidence Editor will enforce this restriction.

The following properties can be set in respect of an Optional Parent Relationship:

*Table 23: Optional Parent Properties*

Property name	Description
Relationship Name	<p>This property sets the Model Relationship Name for the selected Optional Parent Relationship. The value of this property is treated as the internal identifier for the relationship and must be unique across all Relationships in the Dynamic Evidence Type Version. The Relationship Name is never used in the generation of screens for evidence record pages, and so the case worker will never see its value. They are however used in the generation of CER Data and Processing Rule Sets for this Dynamic Evidence Type Version - see <a href="#">Eligibility and entitlement rule sets on page 92</a> for more details.</p> <p>Relationship Names must follow a specific naming format:</p> <ul style="list-style-type: none"><li>• They must start with a lowercase English alphabetic character</li><li>• They can only contain lowercase or uppercase English alphabetic characters, numeric characters and underscores</li><li>• They must not contain any reserved words such 'relatedEmployment' or 'comments' as these are reserved identifier in the Dynamic Evidence Editor.</li></ul>
Parent Type Code	<p>This property setting specifies the Evidence Type Code for the Optional Parent. At runtime, the case worker must select one instance of a evidence record for the Parent Type Code during the creation of the child evidence record.</p>
Description	<p>This property configures a non-localizable Model description value for the selected Optional Parent Relationship. This is for annotative purposes only, and is never displayed to the case worker at runtime.</p>



### ***Deleting attributes***

Administrators can remove attributes of any type from the dynamic evidence type version by clicking **X** in the attribute in the evidence class on the model canvas.

**Note:** Similar to changing the Type of a Data Attribute or Calculated Attribute, if the Attribute being deleted is currently referenced in Validations, Summary Information, Business Start and End Dates, Related CP Properties, or has a Field in respect of it in the User Interface, the Editor will confirm if the administrator still wants to delete them. If the administrator confirms the delete, the Editor will proceed to remove all traces of the deleted Attribute from the Dynamic Evidence Type Version (e.g. by blanking out Business Start and End Dates, by removing Validations which reference it, by removing User Interface components which reference it, etc.)

### ***Deleting relationships***

Administrators can remove relationships of any type from the dynamic evidence type version.

Note that it is not valid to remove a Relationships if the Dynamic Evidence Type in question has an Active Version with this relationship present (the Editor will allow it, but this will fail Activation for the modified Dynamic Evidence Type Version).

This is achieved by clicking the 'x' button in the Mandatory or Optional Parent class on the Model Canvas. Deleting a Relationship will remove the class and its line to the Dynamic Evidence Type Version class from the Model Canvas, trigger all other Relationship classes to reposition themselves on the Model Canvas, and if the last Relationship was deleted, trigger the Dynamic Evidence Type Version class to reposition itself.

### ***Saving dynamic evidence type version updates***

Click **Save** to save the current state of the dynamic evidence type version to the database. Use **Save** for both the **Model** and **User Interface** tabs.

## **Defining the user interface**

Use the **User Interface** tab of the dynamic evidence editor to graphically specify the contents and layout of the evidence record screen **Fields and Clusters**.

### ***The canvas***

The Canvas Panel is used as the drawing surface for the evidence record screens for the dynamic evidence type version.

On this canvas, administrators can create, drag and drop, and maintain the following artefacts:

- Attribute Clusters
- Data Attribute Fields
- Calculated Attribute Fields
- Address Clusters
- Related Participant Clusters
- Related Employment Clusters
- Comments Clusters
- Utility Fields

The Canvas allows the administrator to see the layout of the evidence record screen layout more or less as it will be viewed by the case worker when creating, modifying and reading evidence records in respect of this Dynamic Evidence Type Version.

**Note:** The User Interface Canvas is not an entirely "what you see is what you get" (WYSIWYG) editor. At runtime, the Dynamic Evidence infrastructure generates evidence record screens that are similar to, but not absolutely identical to, the contents of the User Interface Canvas. For example, generated Modify pages for a Dynamic Evidence Type Version will contain the following items which are not drawn in the User Interface Canvas in the Editor:

- Save Button
- Cancel Button
- Asterisks for Mandatory Fields
- Evidence Received Date
- Effective Date of Change
- Change Reason
- Codetable Items for Codetable Data Attributes
- etc.

However, these discrepancies are relatively minor in visual impact, and in general it is easy for the administrator to see how the evidence record screens will look at runtime when using the User Interface Canvas.

Properties for each artefact in the preceding list can be maintained by selecting that component on the Canvas and editing the associated properties in the Properties Panel (which on the User Interface tab is located below the Canvas). The following sections describe each User Interface artefact in detail, describing what the artefact is used for, its properties, and the impact these properties have on the evidence record screens at runtime.

**Mapping model attributes to user interface artefacts**

Two basic kinds of artefact can exist on the User Interface Canvas: fields and clusters.

**Fields**

Fields contain values to be displayed on runtime evidence record screens in respect of dynamic evidence type versions. Fields mainly consist of a label and a value.

Fields are defined in terms of Data Attributes or Calculated Attributes - a Field is basically the Screen representation of such an Attribute, with a number of additional properties which can be set. The Attribute's data type will determine what widget the Field will use at runtime in order to display the Attribute value. The following table shows a mapping between data types and Field widgets used to render these types:

Table 24: Attribute Data Types and their Corresponding Field Renderers

Attribute Data Type	Field Renders Using:
String, Integer, Money, Float	Text Box
Codetable	Dropdown
Boolean	Checkbox
Date	Cúram Date widget
DateTime	Cúram DateTime widget

It is not necessary for the administrator to specify which widget to use; these are automatically inferred by the Dynamic Evidence infrastructure.

### *Reordering fields*

When a field is an attribute cluster on the user interface canvas, a field can be reordered at any time, and can be moved both within attribute clusters and between attribute clusters.

In the first case, it is possible to move a Field from one position in an Attribute Cluster to another position in the same Attribute Cluster. To do this, click on the Label of the Field in question, and, by holding the mouse button down, drag it until it is directly over the Label of the Field with which it is to be positionally swapped. Finally, release the mouse button, and the Field will move to its new position.

**Note:** Clusters cannot be dropped onto the position they current occupy

In the second case, it is possible to move a Field from one Attribute Cluster to another Attribute Cluster. This will have the effect of removing it completely from the first Cluster, and adding it as the last Field in the second Cluster. To do this, click on the Label of the Field in question, and, by holding the mouse button down, drag it until it is directly over the middle of the Attribute Cluster to which it is to be moved. Finally, release the mouse button, and the Field will move to its new position.

### **Clusters**

Clusters are containers of information on evidence record screens, and they come in a number of different varieties.

- Attribute Clusters, which are the containers for Fields. Data Attributes and Calculated Attributes can both be dragged from the accordion view in the Palette and dropped as Fields onto Attribute Clusters
- Address Clusters, which are the User Interface representations of Model Address Attributes. Address Attributes can be dragged from the accordion view in the Palette and dropped as Clusters onto the area over an existing Cluster.
- Related Case Participant Clusters, which are the User Interface representations of Related Case Participant Attributes. Like Address Attributes, Related Case Participant Attributes can be dragged from the accordion view in the Palette and dropped as Clusters onto the area over an existing Cluster.
- Related Employment Clusters, which are the User Interface representations of Related Employment Attributes. Related Employment Attributes can be dragged from the accordion view in the Palette and dropped as Clusters onto the area over an existing Cluster.
- Comments Clusters, the User Interface representations of Model Comments Attributes. Comments Attributes can also be dragged from the accordion view in the Palette and dropped as Clusters onto the area over an existing Cluster.

### *Reordering and deleting clusters*

When clusters are on the User Interface Canvas, clusters can be reordered at any time. This is especially important for attribute clusters as attribute clusters are always added as the last cluster on a page.

Clusters of any type can be reordered in exactly the same manner - this method applies equally to Attribute Clusters, Address Clusters, Related Case Participant Clusters, Related Employment Clusters and Comments Clusters.

To reorder a Cluster, click on a white-colored area of the Cluster to be moved and, by holding the mouse button down, drag it to the area directly over another Cluster. When the Cluster is over an area onto which it can be dropped, the drop area will change color to light blue. Finally, release the mouse button, and the Cluster will move to its new position.

**Deleting clusters**

Administrators can delete clusters from the User Interface Canvas at any time by clicking the 'x' that appears when the mouse is over the cluster in question. The cluster is then immediately removed from the screen, and the page is redrawn.

**Attribute clusters**

To create an attribute cluster, click **Add Attribute Cluster** on the user interface palette. This is the only way to create attribute clusters.

Attribute clusters are the only user interface artefact that you can create by clucking a button. The attribute cluster is added as the last cluster in the user interface canvas. Administrators can, if required, move the newly-added attribute cluster to another location on the canvas.

At Activate time, Dynamic Evidence Type Versions will be checked to ensure that all Attribute Clusters contain at least one Field.

The following properties can be set in respect of a Data Attribute Cluster:

Table 25: Standard Cluster Properties

Property name	Description
Title	An optional Cluster Title for the default locale. Note that most Clusters have Titles, but they are not in fact enforced as mandatory by the Dynamic Evidence infrastructure.
Title ID	A Resource property identifier for the Title property. Mandatory if a Title is provided (this property is only displayed in the Editor if the Title has one or more characters).
Description	An optional Cluster Description for the default locale. Cluster Descriptions, if provided, appear after the Title at runtime.
Description ID	A Resource property identifier for the Description property. Mandatory if a Description is provided (this property is only displayed in the Editor if the Title has one or more characters).
Number of Columns	A mandatory numeric drop down containing values from 1 to 4, the Number of Columns dictates how many Columns will be used to lay out all contained Fields in this Attribute Cluster in generated create, modify and view pages in respect of this Dynamic Evidence Type Version. Typically, this value is set to 2 (and occasionally 1) for most Dynamic Evidence Types.
Label Width	The percentage of the width of an Attribute Cluster column that the label should occupy. The specified width will be applied to every other column. In most cases this should be set to 40 which is a default value.
On Create Page	Possible values are true or false. If true, the Attribute Cluster is shown on the evidence create page; if false, the Cluster is not displayed on the Create page. In most cases this should be set to true (i.e. The check box should be checked).

Property name	Description
On Modify Page	Possible values are true or false. If true, the Attribute Cluster is shown on the evidence modify page; if false, the Cluster is not displayed on the Modify page. In most cases this should be set to true (i.e. The check box should be checked).
On View Page	Possible values are true or false. If true, the Attribute Cluster is shown on the evidence view page; if false, the Cluster is not displayed on the View page. In most cases this should be set to true (i.e. The check box should be checked).
Online Help	An optional property containing Cluster information for Online Help on evidence record screens.

**warning** Care must be taken with the use of the **On Create** page and **On Modify** page properties to ensure that Clusters which do not appear on evidence maintenance screens do not contain Fields in respect of Data Attributes which are marked as Mandatory. This is a logical error (which is not currently enforced by the Editor), in that it would not be possible for case workers to provide values for mandatory Attributes, and hence should be avoided.

### **Data attribute fields**

You can drag data attributes from the accordion control on the user interface palette and drop the data attributes as fields onto attribute clusters.

The following Field properties can be set for Data Attributes:

Table 26: Data Attribute Field Properties

Property name	Description
Label	Field Label for the default locale is enforced as mandatory by the Dynamic Evidence infrastructure.
Label ID	A Resource property identifier for the Label property. Mandatory if a Label is provided (this property is only displayed in the Editor if the Label has one or more characters).
Source Attribute	The name of the Model Data Attribute that this Field populates.
Modifiable	Possible values are true or false. This property determines whether or not the Field is read-only on the evidence modify page. If false (i.e. The Field is read-only after first creation), then the Field Value is rendered as a Label. Otherwise an editable widget appropriate to the associated Data Attribute Type will be used.
Use Default	Possible values are true or false. A true setting indicates that the field should use the default value specified for the <code>Default Value</code> property for the related Source Attribute. The case worker user will see the Default Value populated in this field on an evidence record create operation. If no Default Value has been provided in the Data Attribute Model Properties, but Use Default is true, then the Field will be populated with appropriate default values for the data type in question (e.g. the current date for Data Attributes of type <code>Date</code> , 0 for Data Attributes of type <code>Integer</code> , etc).

Property name	Description
Use Blank	This property is only applicable if the Source Attribute has a data type of <code>Codetable</code> . This property indicates the dropdown for the Source Field should have a selectable blank value in it (in effect, making the Codetable mandatory or non-mandatory). It has a setting of true or false. If true the caseworker will see a blank value added to the list of possible selectable Codetable values in the dropdown for this Field. If it is set to false, only the list of Codetable values will be displayed, thus making the Field mandatory.
Online Help	An optional localizable text containing field or attribute information for Online Help on evidence record screens.

### **Calculated attribute fields**

You can drag calculated attributes from the accordion control on the user interface palette and drop the calculated attributes as fields onto attribute clusters.

The following Field properties can be set for Calculated Attributes:

*Table 27: Calculated Attribute Field Properties*

Property name	Description
Label	An optional Field Label for the default locale. Note that most Fields have Labels, but they are not in fact enforced as mandatory by the Dynamic Evidence infrastructure.
Label ID	A Resource property identifier for the Label property. Mandatory if a Label is provided (this property is only displayed in the Editor if the Label has one or more characters)
Source Attribute	The name of the Model Calculated Attribute that this Field references.
Online Help	An optional localizable text containing calculated attribute information for Online Help on evidence record screens.

### **Address clusters**

You can drag address attributes from the accordion control on the user interface palette and drop the data attributes over existing clusters as address clusters.

The property settings for address clusters are the same as those for attribute clusters. For more information, see [Attribute clusters on page 68](#).

### **Related case participant clusters**

To configure properties that determine how case workers will interact with the related case participants are participants.

Related Case Participants are Participants other than the Primary Client to be associated with an evidence record. Related Case Participants allow case workers to do one of three things:

- Select an existing Case Participant (i.e. a Participant who has already been added to the Case) to associate with the current an evidence record
- Search for a Participant (i.e. a Participant who has not already been added to the Case) to associate with the current evidence record (and also with the Case, creating a new Case Participant record at the same time)

- Register a new Case Participant of type Representative and associate them with the current evidence record, creating a new Case Participant record at the same time

Related Case Participant Clusters are the User Interface representation of Related Case Participant Attributes.

The following properties can be set for a Related Case Participant Cluster

Table 28: Related Case Participant Cluster Properties

Property name	Description
Name for Cluster	An optional (but practically essential) localizable partial name for the Related Case Participant Cluster (partial, as whatever the administrator adds as text in this field will be suffixed by Details e.g. <b>Income Details</b> ).
Name for Cluster ID	A Resource property identifier for the Name for Cluster property. Mandatory if a Name for Cluster is provided (this property is only displayed in the Editor if the Name for Cluster has one or more characters).
Name for Labels	An optional (but again practically essential) localizable partial name to use for a number of Labels on the 3-Panel Related Case Participant Cluster. These are: <ul style="list-style-type: none"> <li>• For the first Panel, the field label will be set to <b>Name for Labels Value Participant</b></li> <li>• For the second Panel, the field label will be set to <b>Name for Labels Value</b></li> <li>• If the <b>One Name</b> field is set to true, the field label for the first field in the third Panel will be set to <b>Name for Labels Name</b></li> <li>• If the <b>One Name</b> field is set to false, the field label of the first field in the third Panel will be set to <b>Name for Labels First Name</b>, and the field label for the second field in the third Panel will be set to <b>Name for Labels Surname</b>.</li> </ul>
Name for Labels ID	A Resource property identifier for the Name for Labels property. Mandatory if a Name for Labels is provided (this property is only displayed in the Editor if the Name for Labels has one or more characters).
Name for Descriptions	An optional (but again practically essential) localizable Label to be used in all descriptions on the Related Case Participant Panel (e.g. If the <b>Name for Descriptions</b> is a case participant, please select from the following list of case participant).
Name for Descriptions ID	A Resource property identifier for the Name for Descriptions property. Mandatory if a Name for Descriptions is provided (this property is only displayed in the Editor if the Name for Descriptions has one or more characters).
Case Participant Descriptor	The setting of this value will be used in the label of the Case Participant Panel(the first panel) as a suffix to the in the For Example <b>Name for Labels Value Case Participant Descriptor</b> The Default value is Participant.
Case Participant Descriptor ID	A Resource property identifier for the Case Participant Descriptor property. Mandatory if a Case Participant Descriptor is provided (this property is only displayed in the Editor if the Case Participant Descriptor has one or more characters).



Property name	Description
First Name Label	This value setting is used as an alternative label setting for the First Name field label of the Related Case Participant widget Third Panel. The First Name field value will be set to this property setting if it has 1 or more characters .
First Name Label ID	A Resource property identifier for the First Name Label property. Mandatory if a First Name Label is provided (this property is only displayed in the Editor if the First Name Label has one or more characters).
Second Name Label	This value setting is used as an alternative label setting for the Second Name field label of the Related Case Participant widget Third Panel. The Second Name field value will be set to this property setting if it has 1 or more characters. This property is not displayed if the <b>One Name</b> field setting is true.
Second Name Label ID	A Resource property identifier for the Second Name Label property. Mandatory if a Second Name Label is provided (this property is only displayed in the Editor if the Second Name Label has one or more characters). This property is not displayed if the <b>One Name</b> field setting is true.
Allow Modification	Possible values are <b>no</b> , <b>single</b> , and <b>multiple</b> . The functionality of this setting is described in conjunction with the <b>Show all Panels</b> property. Please refer to <a href="#">Show All Panels and Allow Modification on page 73</a> for more information.
Search Type	The <b>Search Type</b> property determines the search popup page that the case worker will see on create and modify pages. This property has such values as Person, Employer, Product Provider, and Service Supplier, its default setting being blank. E.g. If <b>Search Type</b> is <b>Person</b> , the caseworker will see a Person search popup when searching for related participants. If blank, the search widget seen on the create/modify pages will be a multi popup search type, where the type of search result can be specified by the case worker at runtime.
Show All Panels	Possible values are true or false. Please refer to <a href="#">Show All Panels and Allow Modification on page 73</a> .
Field	Possible values are true or false. If this option is set to true, a single name field (called <b>Name</b> ) will be presented for any new Related Case Participant created. If this value is false then both First and Second Name can be specified.
On Create Page	Possible values are true or false. If true, the Related Case Participant Cluster is shown on the evidence create page; if false, the Cluster is not displayed on the Create page. In most cases this should be set to true (i.e. The checkbox should be checked).
On Modify Page	Possible values are true or false. If true, the Related Case Participant Cluster is shown on the evidence modify page; if false, the Cluster is not displayed on the Modify page. In most cases this should be set to true (i.e. The checkbox should be checked).
On View Page	Possible values are true or false. If true, the Related Case Participant Cluster is shown on the evidence view page; if false, the Cluster is not displayed on the View page. In most cases this should be set to true (i.e. The checkbox should be checked).



Property name	Description
Default To Blank	<p>Possible values are true or false. If true, the Participant drop-down defaults to blank option on Related Case Participant cluster when evidence is being added. If false, the Participant drop-down defaults to the primary case participant if this participant is available in the drop-down.</p> <p>This option has no effect on what is displayed on evidence modify screen, as when evidence is modified the entry displayed represents existing case participant value. It may be empty if no case participant was specified upon evidence creation.</p>
Online Help	An optional localizable text containing Cluster information for Online Help on evidence record screens.
Enable Multi Case Member Updates	<p>This attribute, when set will allow updates to evidence to be applied across active case participants. This property is only available in the participant that is selected in the <b>Related CP Attribute</b> in the <b>General Properties</b>. To enable this feature deselect the <b>Show All Panels</b>. Once selected <b>Allow Modification</b> is set to No.</p>

### Show All Panels and Allow Modification

If `Show All Panels` is set to true for a related case participant cluster, then three panels are displayed in the evidence create page for the related case participant. If the property is set to false, then only the first panel is displayed.

Panel one allows the user to select an existing Case Participant, to associate it with the evidence record currently being recorded.

Panel two provides the user with the ability to search for a Participant on the System.

Panel three presents the user with name, address and phone details fields to register a new participant (of type Representative).

The **Allow Modification** property value governs whether or not a Related Case Participant can be modified.

If **Allow Modification** is set to **multiple**, the case worker can always update the Related Case Participant Reference on the evidence modify page.

If **Allow Modification** is set to **single**, if the Related Case Participant was not entered during the initial evidence record creation, then it will be modifiable on the evidence modify page. However, once the Related Case Participant has been set for the first time and saved (either as a result of a create or modify action), the current Related Case Participant name and age will subsequently be displayed in a label, and no further modification will be possible.

If **Allow Modification** is set to **no**, then the Related Case Participant value is mandatory on create and it cannot be modified subsequently. In this case a read-only **name and age** label for the Related Case Participant is always displayed during in the modify page.

If **Show All Panels** is set to false and **Allow Modification** is set to **no**, the mandatory marker is displayed on the create page

### ***Related employment clusters***

The related employment cluster provides a container for related employment attributes.

A related employment, when dragged on to the evidence canvas, adds a cluster containing fields such as participant and employer. At run time, to create or modify an evidence record, the case worker will be shown a wizard, which mandates that the case worker select an employment record from a list before being able to create the Evidence record. Based on the employment record selection in the wizard, the Related Employment cluster displays its associated Participant Name and Employer Name in the second page of the wizard.

*Table 29: Related Employment Cluster Properties*

Property name	Description
Title	An optional Cluster Title for the default locale. Note that most Clusters have Titles, but they are not in fact enforced to be mandatory by the Dynamic Evidence infrastructure.
Title ID	A Resource property identifier for the Title property. Mandatory if a Title is provided (this property is only displayed in the Editor if the Title has one or more characters).
Description	An optional Cluster Description for the default locale. Cluster Descriptions, if provided, appear after the Title at runtime.
Description ID	A Resource property identifier for the Description property. Mandatory if a Description is provided (this property is only displayed in the Editor if the Title has one or more characters).
Number of Columns	A mandatory numeric dropdown containing values from 1 to 4, the Number of Columns dictates how Columns will be used to lay out all contained Fields in this Related Employment Cluster in generated create, modify and view pages in respect of this Dynamic Evidence Type Version. Typically, this value is set to 2 (and sometimes 1) for most Dynamic Evidence Types.
Related Employment Attribute	The name of the Model Related Employment Attribute that this Field populates.
On Create Page	Non Editable property which defaults to true, which displays the Related Employment Cluster on the Evidence Create Page.
On Modify Page	Possible values are true or false. If true, the Related Employment Cluster is shown on the Evidence Modify page; if false, the Cluster is not displayed on the Modify page. In most cases this should be set to true (i.e. The checkbox should be checked).
On View Page	Possible values are true or false. If true, the Related Employment Cluster is shown on the Evidence View page; if false, the Cluster is not displayed on the View page. In most cases this should be set to true (i.e. The checkbox should be checked).
Online Help	An optional localizable text containing Cluster information for Online Help on evidence record screens.

### Comments clusters

You can drag comments attributes from the accordion control on the user interface palette and dropped the comments attributes over existing clusters as comments clusters. Comments clusters are drawn as a cluster with a single standard multi-line text box.

Table 30: Comments Cluster Properties

Property name	Description
Title	An optional Cluster Title for the default locale. Note that most Clusters have Titles, but they are not in fact enforced to be mandatory by the Dynamic Evidence infrastructure.
Title ID	A Resource property identifier for the Title property. Mandatory if a Title is provided (this property is only displayed in the Editor if the Title has one or more characters).
Description	An optional Cluster Description for the default locale. Cluster Descriptions, if provided, appear after the Title at runtime.
Description ID	A Resource property identifier for the Description property. Mandatory if a Description is provided (this property is only displayed in the Editor if the Title has one or more characters).
Number of Columns	A mandatory numeric dropdown containing values from 1 to 4, the Number of Columns dictates how Columns will be used to lay out all contained Fields in this Comments Cluster in generated create, modify and view pages in respect of this Dynamic Evidence Type Version. Typically, this value is set to 2 (and sometimes 1) for most Dynamic Evidence Types.
On Create Page	Possible values are true or false. If true the Comments Cluster is shown on the evidence create page; if false, the Cluster is not displayed on the Create page. In most cases this should be set to true (i.e. The checkbox should be checked).
On Modify Page	Possible values are true or false. If true the Comments Cluster is shown on the evidence modify page; if false, the Cluster is not displayed on the Modify page. In most cases this should be set to true (i.e. The checkbox should be checked).
On View Page	Possible values are true or false. If true the Comments Cluster is shown on the evidence view page; if false, the Cluster is not displayed on the View page. In most cases this should be set to true (i.e. The checkbox should be checked).
Online Help	An optional localizable text containing Comments Cluster information for Online Help on evidence record screens.

### Utility fields: Skip and Label fields

You can drag the Skip and Label fields from the user interface palette and drop the fields onto attribute clusters.

#### Skip fields

The Skip Field causes a blank field to be drawn in the position specified in its location in the Attribute Cluster. Skip Fields allow for greater control over the layout of the fields in the containing Cluster. Skip Fields have no properties.

**Label fields**

The Label Field causes text information to be drawn in the position specified in its location in the Attribute Cluster. Label Fields allows adding text information to dynamic evidence screens. It has the ability to show or hide the text information on create,modify or view dynamic evidence screens.

Following properties can be set in respect of Label Field.

Table 31: Label Field Properties

Property name	Description
Label	Field Label for the default locale is enforced as mandatory by the Dynamic Evidence infrastructure.
Label ID	A Resource property identifier for the Label Field property. Mandatory if a Label is provided (this property is only displayed in the Editor if the Label has one or more characters).
On Create Page	Possible values are true or false. If true, the Label Field in an Attribute Cluster is shown on the evidence create page; if false, the Label Field is not displayed on the Create page. In most cases this should be set to true (i.e. The check box should be checked).
On Modify Page	Possible values are true or false. If true, the Label Field in a Attribute Cluster is shown on the evidence modify page; if false, the Label Field is not displayed on the Modify page. In most cases this should be set to true (i.e. The check box should be checked).
On View Page	Possible values are true or false. If true, the Label Field in an Attribute Cluster is shown on the evidence view page; if false, the Label Field is not displayed on the View page. In most cases this should be set to true (i.e. The check box should be checked).
Online Help	An optional localizable text containing label field information for Online Help on evidence screens.

**Skip Fields**

The Skip Field causes a blank field to be drawn in the position specified in its location in the Attribute Cluster. Skip Fields allow for greater control over the layout of the fields in the containing Cluster. Skip Fields have no properties.

**Label Fields**

The Label Field causes text information to be drawn in the position specified in its location in the Attribute Cluster. Label Fields allows adding text information to dynamic evidence screens. It has the ability to show or hide the text information on create,modify or view dynamic evidence screens.

Following properties can be set in respect of Label Field.

Table 32: Label Field Properties

Property name	Description
Label	Field Label for the default locale is enforced as mandatory by the Dynamic Evidence infrastructure.

Property name	Description
Label ID	A Resource property identifier for the Label Field property. Mandatory if a Label is provided (this property is only displayed in the Editor if the Label has one or more characters).
On Create Page	Possible values are true or false. If true, the Label Field in an Attribute Cluster is shown on the evidence record create page; if false, the Label Field is not displayed on the Create page. In most cases this should be set to true (i.e. The check box should be checked).
On Modify Page	Possible values are true or false. If true, the Label Field in a Attribute Cluster is shown on the evidence record modify page; if false, the Label Field is not displayed on the Modify page. In most cases this should be set to true (i.e. The check box should be checked).
On View Page	Possible values are true or false. If true, the Label Field in an Attribute Cluster is shown on the evidence record view page; if false, the Label Field is not displayed on the View page. In most cases this should be set to true (i.e. The check box should be checked).
Online Help	An optional localizable text containing label field information for Online Help on evidence record screens.

## Configuring the multiple participant evidence update

Administrators can use the dynamic evidence editor to configure the multiple participant evidence update.

The dynamic evidence editor is used to enable the multiple participant evidence update. To configure the multiple participant evidence update, the developer must perform the following steps;

1. Set the participant in the Related CP Attribute drop down in the **General Properties** in the Dynamic Evidence Editor. For more information, refer to the **Related CP Attribute** entry in the [General Properties on page 34](#) section
2. Deselect the **Show All Panels** in the related participant properties panel and select **Enable Multi Case Member Updates** in the related participant properties panel. For more information refer to the **Enable Multi Case Member Updates** entry in the [Related case participant clusters on page 70](#) section.

## Dynamic evidence rule sets

You can use CER rule sets with dynamic evidence for specific purposes.

### **Generated Rule Sets**

Rule Sets that are written for calculated attributes, summary information and validation are collectively referred to as containing evidence processing logic.

These Rule Sets need to access data pertaining to the relevant dynamic evidence type as part of their logic. To facilitate this, for every dynamic evidence type a processing Rule Set is generated.

Rule Sets written for determining Eligibility and Entitlement for a particular Program are called Eligibility and Entitlement Determination Rule Sets. These Rule Classes need to access data from Dynamic Evidence Types which are involved in the determination. To facilitate Dynamic

Evidence Types participating in Eligibility and Entitlement determination, a Rule Set called a Data Rule Set is also generated for each Dynamic Evidence Type

These Rule Sets are generated whenever changes to an "In-Edit" Evidence Type Version are saved (note that these Rule Sets are generated per-Dynamic Evidence Type, not per-Dynamic Evidence Type Version).

The following sections discuss the structure of these Generated Rule Sets in detail.

### Processing rule sets

Processing rule sets are generated with a Rule Set name of <Dynamic Evidence Type Logical Name>RuleSet.

The rule set contains a rule class with the name <Dynamic Evidence Type Logical Name> and is associated with the dynamic evidence: processing category. An instance of this rule class represents an evidence record of this dynamic evidence type.

### Data attributes

A rule attribute is generated for each data attribute in the corresponding dynamic evidence type version.

The name of the rule attribute is the same as that of the data attribute. The type of the rule attribute depends on that of the data attribute, but this is not a one-to-one mapping. The following table summarizes the mapping from data attribute types to rule attribute types.

Table 33: Data Attribute Type to Rule Attribute Type Mapping

Data Attribute Type	Rule Attribute Type
Boolean	java.lang.Boolean
Date	curam.util.type.Date
Time	curam.util.type.DateTime
Integer	java.lang.Number
Float	java.lang.Number
Money	java.lang.Number
String	java.lang.String
Codetable	Inbuilt Codetable type in CER

### Related case participant attributes

Each related case participant attribute that is defined for a dynamic evidence type version generates two rule attributes in the processing rule class.

One attribute is of type `java.lang.Number` and is named after the related case participant attribute. The other rule attribute is generated to represent the case participant role object that corresponds to the related case participant attribute. The attribute is named `related_<Related Case Participant Attribute Name>`. The type of this attribute is specified as the `CaseParticipantRole` rule class in `CaseEntitiesRuleSet`.

### Related employment attributes

Any related employment attribute that is defined for a dynamic evidence type version generates two rule attributes in the processing rule class.

One attribute is of type `java.lang.Number` and is named after the related employment attribute.

The other rule attribute is generated to represent the employment object that corresponds to the related employment attribute. The attribute is named `related_<Related Employment Attribute Name>` and the type of this attribute is specified as the `Employment` rule class in `ParticipantEntitiesRuleSet`.

### *Parent-child relationships*

The dynamic evidence editor supports the definition of mandatory and optional parents in the modeling section.

Relationships are defined in one of the Dynamic Evidence Type Versions of the Child Evidence Type. As soon as the metadata of the Child Evidence Type Version is saved, a Rule Attribute will be generated. This Rule Attribute can be used to navigate to the Parent Evidence records for a given Child Evidence Record. The name of this Rule Attribute will be the same as that of the Parent Relationship and the type will be `java.util.List<Parent Rule Class>`. In the case where a Dynamic Evidence Type has multiple Parent Types, an Attribute will be generated for each Parent Type.

The Rule Set of the Parent Evidence Type will be updated to contain a Rule Attribute to navigate to the Child Records for a given Parent Record. One such Attribute will be generated for each Child Evidence Type. The Attribute for a particular Child Evidence Type will be generated only when the Child Evidence Type Version containing the Parent definition is activated.

When parent and child relationship is established and both parent and child Evidence Type Versions are activated the system generates a relationship attribute in the parent processing rule class. This attribute is named '**related\_<child Evidence Type logical name>**' and may be used in customized rulesets to navigate from a parent evidence record to its children. This attribute should not be referenced in customized rulesets before both parent and child Evidence Type Versions are activated. If used earlier it will cause a CER validation error when activating the parent or child version. This is not a defect but a recommended approach to developing rulesets for parent child Dynamic Evidence Types.

### *Address attributes*

Each address attribute defined for a dynamic evidence type generates two rule attributes in the processing rule class.

One attribute is of type `java.lang.Number` and is named after the address attribute (this represents the ID of the address).

Another rule attribute is generated to represent the address object that corresponds to the address attribute. The attribute is named `related_<Address Attribute Name>`. The type is specified as the address rule class in `ParticipantEntitiesRuleSet`.

### *Calculated attributes*

No rule attributes are generated for calculated attributes.

Rule attributes that correspond to calculated attributes must be defined in the calculated attributes rule set for this dynamic evidence type version.

### **Data rule sets**

Data rule sets are generated with a rule set name of `<DynamicEvidence Type Logical Name>DataRuleSet`.

The Rule Set will contain a Rule Class whose name is `<Dynamic Evidence Type Logical Name>`, and will be associated with the Dynamic Evidence: Data category. An instance of this Rule Class represents an Active Succession Set of Evidence Records corresponding to this Dynamic Evidence Type.



## Business dates

Business start and end dates are defined in the modeling section of a dynamic evidence type. These dates are used to define the temporal boundaries of a Succession Set. The period covered by these dates is called the *Evidence Business Object Lifetime*. The Editor allows for the mapping of Data Attributes of type `Date` to the Business Start and End Date Attributes of the Dynamic Evidence Type Version.

The Cúram Rule Object Propagation mechanism requires all Data Rule Sets to specify the Evidence Object Lifetime. For Dynamic Evidence Types, the generated Data Rule Set will contain the necessary elements to map the Business Start and End Dates defined in the Editor to the Evidence Lifetime.

## Data attributes

A rule attribute is generated for each data attribute in the corresponding dynamic evidence type version. The name of the Rule Attribute is the same as that of the Data Attribute. The type of the Rule Attribute is dependent on that of the Data Attribute, but this is not a one to one mapping. The table below summarizes the mapping from Data Attribute Types to Rule Attribute Types.

Table 34: Data Attribute Type to Rule Attribute Type Mapping

Data Attribute Type	Rule Attribute Type
Boolean	<code>java.lang.Boolean</code>
Date	<code>curam.util.type.Date</code>
Time	<code>curam.util.type.DateTime</code>
Integer	<code>java.lang.Number</code>
Float	<code>java.lang.Number</code>
Money	<code>java.lang.Number</code>
String	<code>java.lang.String</code>
Codetable	Inbuilt Codetable type in CER

Also, for Data Rule Sets the "Volatile" setting of the Data Attributes are also considered when determining the Data Type of a Rule Attribute. A Data Attribute which is marked as Volatile would have the type determined as described above, but inside a Timeline. This is to facilitate the Rule Object Propagation mechanism to create a Timeline of values while populating each Attribute.

## Parent-child relationship

Like the processing rule set, the data rule set also contains attributes to navigate parent-child relationships. However, this differs from the Processing Rule Set in the following three ways:

- The data type of the parent navigation Attributes will be set to the Data Rule Class for the Parent Evidence Type, rather than a list of Parent Data Rule Class Objects. This is because the Parent Data Rule Class represents a Succession Set and not a single record, and there can only be one Succession Set for each Parent Evidence Type.
- The data type of the child navigation Attribute will be a `java.util.List` of Data Rule Class objects for the Child Evidence Type, as multiple Succession Sets of Child Evidence Types can be related to one Parent Succession Set.



- When parent and child relationship is established and both parent and child Evidence Type Versions are activated the system generates a relationship attribute in the parent processing rule class. This attribute is named `related_<child Evidence Type logical name>` and may be used in customized rulesets (e.g. eligibility and entitlement ruleset) to navigate from a parent evidence record to its children. This attribute should not be referenced in customized rulesets before both parent and child Evidence Type Versions are activated. If used earlier it will cause a CER validation error when activating the parent or child version. This is not a defect but a recommended approach to developing rulesets for parent child Dynamic Evidence Types.

### **Related employment attributes**

The generation logic for related employment attributes is similar to that for processing rule sets.

### **Related case participant attributes**

The generation logic for related case participant attributes is similar to that for processing rule sets, except that for volatile related case participant attributes the data type of both the case participant role ID attribute and the case participant role object attribute are wrapped inside a timeline.

### **Address attributes**

Address attributes are again treated similar to how they are handled for processing rule sets, with the difference that for volatile address attributes, the generated rule attributes will have their data types wrapped inside a timeline.

### **Calculated attributes**

Calculated attributes do not have any corresponding elements generated in the data rule set.

### **Propagator configurations**

The generated data rule sets will be used to access evidence record data from eligibility and entitlement determination rule sets. as such, the data rule sets are useful only if evidence record data is propagated to them at runtime.

Rule Object Propagators are configured to specify the Rule Sets to which data in respect of each Evidence Type should be propagated. As part of activating a Dynamic Evidence Type, a Propagator Configuration is also generated in addition to the Rule Sets. This Propagator Configuration will be named, `<Data Rule Set Name> - Active Succession Set Propagator Configuration`, and will specify that Evidence records of this Dynamic Evidence Type should be propagated to instances of the generated Data Rule Class.

### ***Support for multiple dynamic evidence type versions***

A dynamic evidence type can have multiple dynamic evidence type versions over a period of time.

In each dynamic evidence type version, new attributes and relationships might have been added. Also, existing attributes and relationships might have been dropped. When it comes to representing these variations in the generated rule sets, the two following steps can be applied:

- Generate one rule set that represents all dynamic evidence type versions, that is, one rule set for each dynamic evidence type.
- Generate one rule set for each dynamic evidence type version.

For various reasons, the first option is being used for generating rule sets for dynamic evidence. As such, the generated rule set for a dynamic evidence type contains rule attributes corresponding to the total of all attributes and relationships in respect of all dynamic evidence type versions. For this reason, logic in handcrafted rule sets for summary information or validation or calculated attributes might need to branch based on the availability of a value for an attribute on the evidence record, that is, the rule object that is being processed. The reason is an attribute might have been added from a particular time, and before this time it has no value.

### ***Loading of dynamic evidence rule objects***

Handcrafted rule sets for evidence processing are called from appropriate points during the maintenance of evidence records in respect of dynamic evidence types.

For example, the rule set for summary information is called every time an evidence record for a dynamic evidence type is viewed. The processing rule sets act on a particular evidence record. Dynamic evidence maintenance creates and populates a rule object with the details of the evidence record for which the processing rule set is started.

When the processing rule set is started as part of reading an evidence record, the rule object is populated with the data that is stored in the database. When the processing rule set is started during creation or modification of an evidence record, the corresponding rule object that is created is populated with the new or modified data.

The steps that are involved in loading a dynamic evidence rule object are as follows:

- The generated rule set or rule class for the dynamic evidence type is identified. A rule object is created for this rule class.
- `evidenceID`, `correctionSetID`, `successionID`, `caseID`, `type`, `receivedDate`, `effectiveFrom`, and `status` are attributes that are common to all dynamic evidence types and are populated from the Evidence Descriptor.
- Each rule attribute corresponding to the data attributes are populated with values from the database, or the value that is specified by the user if the evidence record is being created or modified.
- If the evidence record is being created:
  - `evidenceID`, `correctionSetID`, and `successionID` are all set to zero
  - Rule attributes for navigating to parent records that are populated with the parent records selected as part of the create process. Rule attributes navigate in this way because the normal derivation of these rule attributes tries to read the parent records from the database and the relationship is not yet established as the evidence record is yet to be created. Also, rule attributes corresponding to child evidence types would be populated with an empty list for this evidence instance.
  - The rule attribute for address attributes has a value of zero. The rule attribute corresponding to the address object are populated with the address details that are entered during creation.
  - The rule attribute for related case participant attributes has a value of zero, if an existing case participant is not chosen by the caseworker. The rule attribute corresponding to the related case participant object would be populated with the details that are entered during creation.

## Utility rule classes and functions

Some utility java static operations are provided to help with the writing of dynamic evidence processing rule sets.

These Java operations are contained in the `curam.dynamicEvidence.cer.impl.DynamicEvidenceStatics` class. The following section provides a high level description of these operations. Please refer the Javadoc for Dynamic Evidence to obtain more detailed information such as parameters and return types of each operation.

## Listing parent rule objects

For each parent dynamic evidence type, a rule attribute is available in the generated processing rule class.

This attribute will provide a list of all Parent Records. The operations provided in the `DynamicEvidenceStatics` class to get a filtered list of Parent Evidence records are:

- `getActiveParentList()` - This operation returns a list of Parent Records which are Active.
- `getActiveandPendingChangesParentList()` - This Operation returns a list of In-Edit and Active Parent Records which do not have any pending update. For example, if there are three records R1, R2 and R3 where R1 is active, R2 is active and R3 is an In-Edit correction on R2, this operation would return R1 and R3.
- `getParent()` - This operation returns a Parent Record for the given Dynamic Evidence. Consider the following rules snippet which illustrates the usage:

```
<Attribute name="relatedMember">
  <type>
    <ruleclass
      name="Enrollment"
      ruleset="EnrollmentRuleSet"
    />
  </type>
  <derivation>
    <call
      class="curam.dynamicEvidence.cer.impl.DynamicEvidenceStatics"
      method="getParent"
    >
      <type>
        <ruleclass
          name="Enrollment"
          ruleset="EnrollmentRuleSet"
        />
      </type>
      <arguments>
        <this/>
        <String value="Enrollment"/>
      </arguments>
    </call>
  </derivation>
</Attribute>
```

In this example, the `getParent` method retrieves the rule Object representation of Enrollment evidence type which is used as a value to the `relatedMember` attribute.

### Listing child rule objects

For each child dynamic evidence type, a rule attribute will be available in the generated processing rule class.

This Attribute will provide a list of all Child Records, irrespective of their state. The operations provided in the `DynamicEvidenceStatics` class to get a filtered list of Child Evidence records are:

- `getActiveChildList()` - This operation returns a list of those Child Records which are Active.
- `getActiveandPendingChangesChildList()` - This Operation returns a list of In-Edit and active Child Records which do not have any pending update.
- `getChildList()` - This operation returns a list of Child Records for the given Dynamic Evidence. Consider the following rules snippet which illustrates the usage:

```
<Attribute name="relatedEmployerSponsoredCoverage">
  <type>
    <javaclass name="java.util.List">
      <ruleclass
        name="EmployerSponsoredCoverage"
        ruleset="EmployerSponsoredCoverageRuleSet"
      />
    </javaclass>
  </type>
  <derivation>
    <call
      class="curam.dynamicEvidence.cer.impl.DynamicEvidenceStatics"
      method="getChildList"
    >
      <type>
        <javaclass name="java.util.List">
          <ruleclass
            name="EmployerSponsoredCoverage"
            ruleset="EmployerSponsoredCoverageRuleSet"
          />
        </javaclass>
      </type>
      <arguments>
        <this/>
        <String value="EmployerSponsoredCoverage"/>
      </arguments>
    </call>
  </derivation>
</Attribute>
```

In this example, the `getChildList` method retrieves the list of `EmployerSponsoredCoverage` evidence type which is used as a value to the `relatedEmployerSponsoredCoverage` attribute.

### Listing rule objects for a particular dynamic evidence type

Occasionally, it is necessary to obtain a list of rule objects for a dynamic evidence type for a particular case. For example, a validation rule set might get a list of rule objects of a particular dynamic evidence type on a particular case, and check them against the evidence record being validated to perform duplicate checks.

Three relevant operations are provided for this:

- `getEvidenceListForCase()` - Returns a list of Rule Objects pertaining to Evidence Records in respect of a particular Dynamic Evidence Type for a particular Case.
- `etActiveEvidenceListForCase()` - Returns a list of Rule Objects pertaining to Active Evidence Records in respect of a particular Dynamic Evidence Type for a particular Case.

- `getActiveAndPendingChangesEvidenceListForCase()` - Returns a list of Rule Objects pertaining to Active and In-Edit Evidence records with no pending updates for a particular Dynamic Evidence Type for a particular Case.

### Attribute availability

A processing rule set can be used for more than one dynamic evidence type versions. For example, the same validation rule set can be used for many versions of an evidence type. In such a case, the validation rule set must factor the structural changes across different versions into its logic.

For example, Version 1 of an evidence type might contain "authorizedExpense" and "actualExpense" as two attributes. In Version 2, the "actualExpense" attribute might be replaced by a child evidence type that is named "Expense". In this case, the total actual expense should be calculated by summing the expense value from all child records of type "Expense". If there is a validation that checks the actual expense amount against the authorized expense amount and the same validation rule set is used for Version 1 and 2, the validation logic first checks for the existence of the "actualExpense" attribute. This is required because the generated rule set contains rule attributes corresponding to both Version 1 and Version 2.

The operation `isAttributeAvailable()` can be used to check if a particular attribute is available in a particular evidence record. So for this example, the validation rule set can check if the "actualExpense" attribute is available in the record that is being validated. If not, the logic can calculate the actual expense by adding up the expense from "Expense" child records.

### Specific dynamic evidence rule set types

CER rule sets in dynamic evidence can be used for summary information rule sets, validation rule sets, calculated attributes rule sets, and eligibility and entitlement rule sets.

for each of these types of dynamic evidence rule set, we discuss the contract expected of the rule set and how to author them.

### Summary information rule sets

Typically, you can use summary mapping functionality for defining summary information for a dynamic evidence type version. However, sometimes the summary information may not be a straight forward mapping to attributes of the dynamic evidence type version. For example, the summary information might need data from a parent or child evidence record. In such cases, CER rule sets can be used to define the summary information.

An administrator can specify the Rule Set to be used by enabling the **Use Rule Set** radio button, and then specifying the Rule Set using the **Rule Set Name** option.

### Contract

Dynamic evidence has certain expectations for the required structure of summary rule sets in terms of the rule classes and attributes that they should contain. However, the generic CER infrastructure does not have any evidence processing concepts.

So in order to make sure handcrafted Summary Rule Sets meet these expectations, the following restrictions are enforced on them:

- Summary Rule Sets must contain one Concrete Rule Class which extends from `DefaultEvidenceSummary` Rule Class in the `EvidenceSummaryRuleSet`.
- This Rule Class must contain a Rule Attribute named "evidence". The type of this Rule Attribute should be the generated Processing Rule Class, and the derivation of this Attribute should use the specified expression. The specified expression is the default derivation when

an Attribute is defined in the CER Editor. So, for example, if the logical name of a Dynamic Evidence Type is "Alien", a Rule Set named "AlienRuleSet" with a Rule Class named "Alien" would be generated. In this case, the concrete Rule Class in the Summary Rule Set for this Dynamic Evidence Type must contain an Attribute named "evidence" whose type is the "Alien" Rule Class in the "AlienRuleSet".

### Authoring

The attributes of the `DefaultEvidenceSummary` rule class correspond to the different elements of summary information.

*Table 35: DefaultEvidenceSummary Attributes*

Rule Attribute	Type	Value
startDate	Date	null
endDate	Date	null
isStartDateAvailable	Boolean	true
isEndDateAvailable	Boolean	true
summary	String	null
participantDetails	String	null

The idea behind defining these Attributes in the `DefaultEvidenceSummary` Rule Class is twofold:

- In their Summary Rule Sets, administrators need to define only those Attributes which need a different value from those defined in `DefaultEvidenceSummary`.
- In a future release of Cúram, if a new element is added to the Summary Information infrastructure, a corresponding Attribute will be added to `DefaultEvidenceSummary`. So any existing hand-crafted Summary Rule Sets would not need to be changed to include this new Attribute, as long as the default value provided in `DefaultEvidenceSummary` is applicable.

Authoring a Summary Rule Set thus involves re-defining (overriding in Object-Oriented terminology) only those Attributes whose values should be different from those defined in `DefaultEvidenceSummary`.

The values from the `startDate` and `endDate` Attributes are used in the "Period" column of the Dynamic Evidence Type Workspace page only if `isStartDateAvailable` and `isEndDateAvailable` have a value of true. Otherwise, the Business Start Date and Business End Date options defined in the *Modeling* section is used to calculate the Evidence Workspace "Period" column. As such, if the intention is not to set `startDate` and `endDate` as part of a Summary Information Rule Set, then the Rule Set should set `isStartDateAvailable` and `isEndDateAvailable` to false.

Typically, the logic to derive different elements of Summary Information is based on the data from the corresponding evidence record instance. This is why the contract for Summary Rule Sets specify that they should have an Attribute named "evidence" -this Attribute is populated with data from the corresponding evidence record instance whenever the Rule Set is invoked.

Dynamic Evidence Administration provides support for authoring Summary Rule Sets by generating a "starter" Rule Set, if the Summary Information Rule Set specified does not exist already. The following sections talk about the process of authoring a Summary Rule Set by using such a "starter" Rule Set, or by reusing an existing Summary Rule Set.

## Using a Starter Rule Set

If the Summary Rule Set specified for a Dynamic Evidence Type Version does not exist, a "starter" Rule Set with the given name is generated. The generated starter Rule Set will have a class called "SummaryInformation" with the base Rule Class and the "evidence" Attribute mentioned in the previous section. This Rule Set will be associated with the Dynamic Evidence Summary Information category.

The Rule Set should be further edited by the administrator to define an Attribute corresponding to each element of Summary Information for which the value defined in `DefaultEvidenceSummary` should be overridden. When the Dynamic Evidence Type Version is activated, the Summary Rule Set is also activated.

## Using an Existing Rule Set

Sometimes, an existing Rule Set is specified as the Rule Set to be used for Summary Information. Typically, this happens when a new Dynamic Evidence Type Version is created and the previous Dynamic Evidence Type Version already had a Rule Set specified for Summary Information.

In this case, the existing Rule Set is not modified automatically. This Rule Set needs to be modified only if the logic to compute the Summary elements needs to change. A point to note is that if an existing Summary Rule Set is modified, the changes will be visible to all the Dynamic Evidence Type Versions which have been using this Rule Set. So, if the changes to the Summary Information derivation is required for only the new Dynamic Evidence Type Version, a new Summary Rule Set should be used instead of modifying an existing Rule Set.

## Validation rule sets

Typically, you can use standard validations for defining validations for an dynamic evidence type version. However, the validations might be a complex combination of many conditions or might involve calculations based on data from related parent or child evidence record instances. In such cases, CER rule sets can be used to define the validations.

An administrator may specify the Rule Set to be used by enabling the **Use Rule Set** radio button under **Additional Validations** in the **Validations** Tab of the Evidence Properties, and then specifying the Rule Set using the **Rule Set Name** option.

### *Contract*

As with summary information, dynamic evidence has certain expectations for the required structure of validation rule sets in terms of the rule classes and attributes that must be contained. However, the generic CER infrastructure does not have any evidence processing concepts.

So, to make sure handcrafted Validation Rule Sets meet these expectations, the following restrictions are enforced on them:

- Validation Rule Sets should contain one concrete Rule Class which extends from the `DefaultEvidenceValidationResult` Rule Class in the `EvidenceValidationRuleSet`.
- This Rule Class must contain a Rule Attribute named "evidence". The type of this Rule Attribute should be the generated Processing Rule Class, and the derivation of this Attribute should use the "specified" expression. The "specified" expression is the default derivation when an Attribute is defined in the CER Editor.

So, for example, if the logical name of a Dynamic Evidence Type is "Alien", a Rule Set named "AlienRuleSet" with a Rule Class named "Alien" will be generated. In this case, the concrete Rule Class in the Validation Rule Set for this Evidence Type must contain an Attribute named "evidence" whose type is the "Alien" Rule Class in the "AlienRuleSet".



Infrastructure rule classes

The `EvidenceValidationRuleSet` provides several rule classes to use when authoring validation rule sets.

Validation

The `Validation` Rule Class represents a particular Validation. It contains the following Attributes:

Table 36: Validation Rule Class - Attributes

Name	Type	Description
<code>isFailure</code>	<code>Boolean</code>	Indicates if the Validation has failed for the evidence record.
<code>failureMessage</code>	<code>curam.creole.value.Message</code>	The message to be shown to the user if the Validation fails.
<code>informationalType</code>	<code>curam.dynamicevidence.validation.impl.InformationalType</code>	Specifies if a Validation failure should be reported as a warning, error or a fatal error. By default, this is set to <code>error</code> . A warning will not stop the user action from getting completed. For example, if a Validation fails during creation of an evidence record, but the informational type is <code>warning</code> , the evidence record will get created in the database and a warning message will be displayed to the user. However, if the informational type is <code>error</code> or <code>fatal error</code> , any database changes as part of the user action will be rolled back. In addition, a fatal error will stop the validation process immediately. The informational type of a Validation should be specified as <code>fatal error</code> , if it is considered that proceeding with subsequent Validations is not useful if this Validation fails.

ValidationMode

The `ValidationMode` Rule Class represents the operation as part of which the Validation is being invoked. This can be used to determine the set of Validations to be applied and the informational type to be used for a particular Validation.

This Rule Class has a single Attribute, namely `mode` of type `curam.dynamicevidence.validation.impl.ValidationMode`.



## DefaultEvidenceValidationResult

As explained in the contract, Validation Rule Sets must have a Rule Class that extends from `DefaultEvidenceValidationResult`. This Rule Class has the following Attributes; all of these Rule Attributes are of type `List<Validation>`

Table 37: *DefaultEvidenceValidationResult* Attributes

Rule Attribute	Purpose	Value
detailsValidations	These Validations are called before writing a new or modified Evidence record to the database. When these Validations are called as part of creating a new evidence record, Parent/Child Relationship will not have been established. As such, any Validation that involves navigating to Parent Evidence records should not be included as part of detailsValidations. Typically these Validations contain Single Field or Cross Field validations.	null
standardValidations	These Validations are called after an evidence record is created, modified, activated or as part of validating an evidence record. Typically this contains Validations which involve navigating to other evidence records such as Parent/Child records or evidence record instances of the same Type on the Case.	null
preCreateValidations	This set of Validations is called before an evidence record and any entity related to that an evidence record are created on the Database. The related entity could be an Address or a Case Participant Role.	null
preModifyValidations	This set of Validations is called before an evidence record and any entity related to that evidence record are modified on the Database. The related entity could be an Address or a Case Participant Role.	null
postCreateValidations	As the name suggests, this set of Validations is called after an evidence record and any entity related to that evidence record are created on the Database. The related entity could be an Address or a Case Participant Role.	null
postModifyValidations	As the name suggests, this set of Validations is called after an evidence record and any entity related to that evidence record are modified on the Database. The related entity could be an Address or a Case Participant Role.	null

In addition to the preceding Attributes, this Rule Class also contains a `validationMode` Attribute.

The idea behind defining these Attributes in the `DefaultEvidenceValidationResult` Rule Class is twofold:

- In their Validation Rule Sets, administrators need to define only those Attributes which correspond to the sets of Validations required for the particular Dynamic Evidence Type Version.
- In a future release of Cúram, if a new set of Validations is added, a corresponding Attribute would be added to `DefaultEvidenceValidationResult`. As such, any existing hand-crafted Validation Rule Sets would not need to be changed to include this new Attribute, as long as the new set of Validations is not required.

### Authoring

To author a validation rule set, define the rule set, define the validation sets, and define a validation.

### Defining the Rule Set

If the Rule Set specified in the "Additional Validations" section does not exist, the Dynamic Evidence infrastructure will generate a "starter" Rule Set. The generated starter Rule Set will have a class called "ValidationResult" with the base Rule Class and an "evidence" Attribute as mentioned in the *Validation Rule Sets Contract* section, and will be associated with the Dynamic Evidence Validation category.

Sometimes, an existing Rule Set is specified as the Rule Set to be used for Validation. Typically, this happens when a new Dynamic Evidence Type Version is created and the previous Dynamic Evidence Type Version already had a Rule Set specified for Validation. If an existing Rule Set is used, it should adhere to the Validation Rule Sets contract. A point to note is that if an existing Validation Rule Set is modified, the changes would be visible to all Dynamic Evidence Type Versions which have been using this Rule Set.

### Defining Validation Sets

As described in the section about `DefaultEvidenceValidationResult` Rule Class, this Rule Class has Attributes corresponding to different sets of Validations. All these Attributes have a value of null, which means by default there are no Validations defined in any of the Validation sets. For a particular Dynamic Evidence Type, all or only some of the Validation Sets might be required. For example, a Dynamic Evidence Type Version might require only "detailsValidations" and "standardValidations". As such, the Attributes corresponding to Validation sets should be redefined. Typically, a CER Fixed List is used to construct the list of Validations in a set.

### Defining a Validation

Typically, a Validation will refer to data from the evidence record or related evidence records - this is why the contract for Validation Rule Sets specify that they should have an Attribute named "evidence". This Attribute is populated with data from the corresponding evidence record instance whenever the Rule Set is invoked.

A Validation can be defined by using the CER create expression. The create expression when called for the `Validation` Rule Class requires that a derivation be specified for the `isFailure` Attribute and `failureMessage` Attribute.

The derivation for the `isFailure` Attribute must produce a boolean value. For example, let us say there are two Attributes, "startDate" and "endDate" in a Dynamic Evidence Type, and the Validation required is that the "startDate" should not be the same as the "endDate". The derivation for the `isFailure` Attribute of this Validation can use a CER compare expression which compares the "startDate" and "endDate" Attributes. The compare expression will obtain the value of these

two Attributes by using a reference expression on the "evidence" Attribute of the Validation Rule Class.

The derivation for the failureMessage Attribute should produce a value of type `curam.creole.value.Message`. For this, the derivation can use an XML Message or a Resource Message expression. Again, if the message is to be parameterized with data from the evidence record, the required Attribute can be accessed through the "evidence" Attribute of the Validation Rule Class using the reference expression.

Optionally, a value can be specified for the "informationalType" of the Validation. If no value is specified, the informational type will be considered in error. However, the static operations ( `error()` , `warning()` and `fatalError()` ) in `curam.dynamicEvidence.validation.impl.InformationalType` can be used to specify a different informational type.

Typically, the informational type is dependent on the Validation Mode. The Validation mode for a particular Validation session can be accessed through the "validationMode" Attribute of the Validation Result Rule Class. The validation mode can then be compared with one or more Validation Modes using the static operations ( `applyChanges()` , `approve()` , `insert()` , `modify()` , `validateChanges()` ) defined in `curam.dynamicEvidence.validation.impl.ValidationMode` to determine the informational type to be used. For example, the Validation Mode for a Validation Session can be compared with "Apply Changes" and "Validate Changes" and the informational type can be specified as error for the former and warning for the latter case.

### **Calculated attributes rule sets**

Calculated attributes are attributes whose value is derived from manipulating values of other attributes of the same dynamic evidence type or related dynamic evidence types.

These Calculated Attributes are typically used in the evidence view pages in respect of a Dynamic Evidence Type, but they could be used for other purposes as well such as in Validations. For example, a Dynamic Evidence Type for Adoption may have Case Participant Role Ids for the Parent and Child; Calculated Attributes could be defined to calculate the names of Parent and Child from the respective Case Participant Role IDs.

For each Dynamic Evidence Type Version, a Calculated Attributes Rule Set must be defined if the Dynamic Evidence Type Version has Calculated Attributes in its model. The following section describes the process of writing a CER Rule Set for Calculated Attributes.

### ***Contract***

Like other dynamic evidence rule set types, dynamic evidence has certain expectations for the required structure of calculated attributes rule sets in terms of the rule classes and attributes that must be contained.

As such, the following restrictions are enforced on such Rule Sets:

- The Calculated Attributes Rule Set must contain one Concrete Rule Class which extends from the DefaultCalculatedAttributes Rule Class in the EvidenceCalculatedAttributesRuleSet Rule Set.
- This Rule Class must contain a Rule Attribute named "evidence". The type of this Rule Attribute must be the generated Processing Rule Class and the derivation of this Attribute should use the "specified" expression. The "specified" expression is the default derivation when an Attribute is defined in the CER Editor. So, for example, if the logical name of a Dynamic Evidence Type is "Alien", a Rule Set named "AlienRuleSet" with a Rule Class named "Alien" will be generated. As such, the concrete Rule Class in the Calculated Rule Set

for this Dynamic Evidence Type must contain an Attribute named "evidence" whose type is the "Alien" Rule Class in the "AlienRuleSet".

- This Rule Class must contain a Rule Attribute corresponding to each Calculated Attribute defined in the Dynamic Evidence Type Version. The name and type of these Rule Attributes must match those of the corresponding Calculated Attribute.

### *Authoring*

While editing the metadata for a dynamic evidence type version, a calculated attributes rule set must be specified that uses the **Calculated Attributes Rule Set Name** option in the evidence properties panel in the evidence editor.

The logic to compute the value of a Calculated Attribute is typically based on the data from the relevant evidence record or from evidence records related to it, and this is why there is a requirement to have an evidence Rule Attribute in Calculated Attribute Rule Sets. When a Calculated Attribute Rule Set is invoked during evidence record maintenance, this Attribute will be populated with a Rule Object containing the data for the evidence record for which the Calculated Attributes have to be computed.

### **Using a Starter Rule Set**

If the specified Rule Set does not exist, a starter Rule Set with the given name will be generated. The generated starter Rule Set will have a class called `CalculatedAttributes` which has the base Rule Class and the evidence Attribute mentioned in the previous section, and will be associated with the Dynamic Evidence Calculated Attributes category.

This Rule Set should be further edited by the administrator to define Attributes corresponding to each Calculated Attribute. When the Dynamic Evidence Type Version is activated, the Calculated Attributes Rule Set is also activated.

### **Using an Existing Rule Set**

Sometimes, an existing Rule Set is specified as the Rule Set to be used for Calculated Attributes. Typically, this happens when a new Dynamic Evidence Type Version is created and the previous Dynamic Evidence Type Version already had a Rule Set specified for Calculated Attributes.

In this case, the existing Rule Set is not modified automatically. If no new Calculated Attributes Rule Set has been added and the logic to compute the Calculated Attributes does not need to change, then the existing Rule Set can be used. If an additional Calculated Attribute is added to the new Dynamic Evidence Type Version, a corresponding Rule Attribute would have to be defined in the existing Calculated Attributes Rule Set.

### **Eligibility and entitlement rule sets**

The eligibility and entitlement determination rule sets are typically architected in three layers.

- **Data Rule Classes**

Data Rule Classes are the closest to Evidence Record Data. They mirror the structure of the Evidence Record Data i.e. they contain Rule Attributes corresponding to Attributes and relationships of a Dynamic Evidence Type. Rule Objects are created for these Rule Classes whenever Evidence records in respect of the corresponding Dynamic Evidence Types are created or modified.

- **Calculator Rule Classes**

Calculator Rule Classes contain calculations which represent more coarse-grained business concepts than those represented by the data elements for Dynamic Evidence Types. For

example, Dynamic Evidence Types representing the different kinds of income and the composition of a Household may exist; A Calculator Rule Class might calculate the Net Household Income by summing up the Income of all the Household members, after applying legislation to exclude those incomes that should not be counted.

Generally, such business concepts are factored in to separate Calculator Rule Classes, so that they can be reused across many Programs. Irrespective of the reuse, factoring big chunks of Rules in to many Calculator classes make the Rule Sets modular and readable.

- **Program Rule Classes**

Program Rule Classes use Data and Calculator Rule Classes to determine the eligibility for a particular Program. These Rule Classes need to exhibit certain characteristics in terms of their structure, so that the Eligibility and Entitlement Engine can work with them to create Determinations. For more details on developing Eligibility and Entitlement Rules, see the *Working with Cúram Express Rules Guide*.

Of these three, Data Rule Classes are generated for all Dynamic Evidence Types. So, for custom Programs, Eligibility and Entitlements Rules can be written by developing Program Rules that make use of these generated Dynamic Evidence Data Rule Classes. Alternatively, Program Rules can be written to make use of a layer of Calculator Rule Classes which in turn make use of the generated Data Rule Classes.

## Localizing dynamic evidence

A description of the artefacts that are involved in the localization process for dynamic evidence, and the steps that are necessary to localize them.

### ***Localizing dynamic evidence process overview***

Localizing dynamic evidence types is similar to localizing other parts of Cúram. The artefacts and mechanisms are broadly the same and only the resource locations and naming conventions are specific to dynamic evidence.

For more information about the localization process, see the *Web Client Reference Manual*.

Most of the following steps involve localizing properties resources in the Application Resource Store. The administration pages for the Resource Store can be accessed as follows:

- Log in as an administrative user, for example, Admin.
- In the Shortcuts panel, expand the **Intelligent Evidence Gathering** section.
- Click **Application Resources**.

### ***Localizing the dynamic evidence editor properties resource in administration suite***

The dynamic evidence editor uses the properties resource

`dynevd_evidenceditor.properties` to specify values in the dynamic evidence editor such as labels, text, tool tip text, labels on button, items in drop downs, messages, and so on.

This properties file contains key/value pairs which are in UTF-8 format. To support localization, locale specific values have to be provided for the keys.

- Using the Application Resource Store administration screens, download the version of the properties resource ( `DynEvd_EvidenceEditor.properties` ) for the default locale.
- Using a text editor, change the value of each property to the appropriate localized equivalent.

- Re-upload the modified properties and publish the changes.
- To view the changes made to properties, launch the Dynamic Evidence Editor. To do this:
  - In the Actions button in the list item, select 'Edit Metadata'

### ***Localizing the runtime dynamic evidence user interface***

Use specific artefacts to localize caseworker runtime evidence record screens, that is, create, modify and view screens, for dynamic evidence types.

#### **Static properties resource**

Dynamic evidence uses a properties resource `dynevdstaticproperties.properties` to store various strings that are common to all dynamic evidence types. For example, the labels for **Save** and **Cancel**, and standard messages used in related case participant clusters. The file must be localized for each desired locale.

To achieve this:

- Using the Application Resource Store administration screens, download the version of `DynEvdStaticProperties.properties` for the default locale.
- Using a text editor, change the value of each property to the appropriate localized equivalent.
- Back in the Administration Suite, create a new Application Resource called `DynEvdStaticProperties.properties` for the desired locale with the newly localized file as the content.

This localization step needs to be performed only once.

#### **Evidence type properties resources**

Dynamic evidence also uses individual properties resources to store evidence type specific localizable text.

There is one such resource per Dynamic Evidence Type and it currently stores only the Evidence Type localizable runtime description. The description property is used in Evidence Maintenance to provide a description for the Dynamic Evidence Type (for example, on the 'New Evidence' screen accessed from the Evidence Dashboard). The naming convention for these properties resources is “`DynEvd_EvidenceType_`” followed by the logical name of the Dynamic Evidence Type and ended with “`.properties`”. They must be localized for each Evidence Type and each supported user locale.

To achieve this:

- Using the Application Resource Store administration screens, download the version of the properties resource ( `DynEvd_EvidenceType_<logicalName>.properties` ) for the default locale.
- Using a text editor, change the value of each property to the appropriate localized equivalent.
- Back in the Administration Suite, create a new Application Resource for the desired locale with the newly localized file as the content.

A new Evidence Type properties resource is created when a new Dynamic Evidence Type is created and it is deleted when the Evidence Type is deleted.

#### **Evidence type version properties resources**

When editing a dynamic evidence type version using the dynamic evidence editor, specific localizable strings are specified in the form of labels and titles for user interface elements.

When the Evidence Type Version is activated, these localizable strings are also stored in properties resources in the Application Resource Store. The naming convention for these



properties resources is `DynEvd` followed by the logical name of the Dynamic Evidence Type and then a numeric form of the Evidence Type Version effective date.

The strings provided via the Dynamic Evidence Editor are used to populate the properties resource for the default locale. This means that if a localized version is not available in a locale-specific properties resource, then the string from the default locale resource will be used via the standard fallback mechanism.

This properties resource must be localized for each Evidence Type Version and locale you wish to support. The localization process is as described in [Static properties resource on page 94](#).

### **Generating the localized user interface**

Specific properties resources must function as the input to other properties files that are generated by the dynamic evidence infrastructure. These properties resources must not be directly referred to by the dynamic evidence case management screens at runtime.

For more information about the property resources that must not be directly referred to by the dynamic evidence case management screens at runtime, see [Evidence type version properties resources on page 94](#).

The actual properties files used at runtime are generated the first time a user accesses the page for a particular locale. Therefore it is important to note that changing the source properties files after this point will not result in any change to the user interface. Ideally, all localization activity should be completed before case workers actually start accessing the Dynamic Evidence screens at runtime.

### ***Message files***

Cúram message files must be localized for each locale that must be supported. The process is the same for dynamic evidence.

### ***Codetable items***

Each dynamic evidence type has a codetable entry generated on the `EvidenceType` codetable.

The codetable description must be correctly localized, as it is used to generate various parts of the user interface, for example, titles for evidence maintenance pages.

## **Dynamic Evidence Configuration Extractor**

Dynamic evidence is an administrative way of defining evidence that generates as many Cúram artefacts as possible. As a result, the learning time and effort that is associated with implementing evidence is reduced.

The Cúram artefacts include dynamic UIM pages (for create, read, modify, and so on), user interface tab configurations, CER rule sets, property files, security identifiers and groups, CER propagators, and so on.

Most customers have Cúram development projects. These customers must locate and export all such generated artefacts so that the artefacts' source code can be controlled. As a result, it is a requirement that configurable artefacts, such as DMX, CTX, clobs, blobs, and so on, can be saved and are available to be used repeatedly. So, when the database is re-created any dynamic evidence types can be persisted and it is not required to re-create the dynamic evidence types each time.

To satisfy this requirement, a Dynamic Evidence Configuration Extractor tool is provided. The tool is responsible for the two following tasks:

- Extracting dynamic evidence configuration information from a runtime Cúram database.
- Writing the information to the file system by using standard Cúram development artefacts, such as DMX, CTX, XML blobs and clobs, tab configuration files, and so on.

### **Features of the Dynamic Evidence Configuration Extractor**

The Dynamic Evidence Configuration Extractor tool extracts Dynamic Evidence configuration information and stores it in standard Cúram development artefacts, for example, DMX, CTX, XML blob/clob and section configuration files. This is done so that these artefacts can be automatically recreated as part of a database rebuild using the existing Cúramdata manager.

Launched using the Cúram batch launcher, the main features of the Dynamic Evidence Configuration Extractor tool are as follows:

- The tool writes its output to a single directory and it expects this to be a standard Cúram component directory (e.g. *custom*). For example, within this directory it creates subdirectories such as *codetable*, *data*, and *tab*.
- Extracting database records with their generated primary keys poses a potential risk for key clashes when the records are uploaded. The reason for this is that when the database is rebuilt, the key generation mechanism is reset and it is very likely to produce the same keys as those in the extracted Dynamic Evidence Types. To avoid this, the extractor replaces generated database primary keys with new keys from a pre-defined key range. The key range is applied to each extracted table individually rather than being shared across all tables (this way keys are used more efficiently). Only two tables: *CreoleRuleset* and *CreoleRulesetEditAction* share the same key range, because both tables are referenced from the same field in *CreoleRulesetCategoryLink* table.
- Tab configuration files (for generated Evidence tabs) are extracted as blobs, as part of extracting the *AppResource* entity. Section configuration files, however, cannot be extracted as individual blobs. They are extracted as contribution section files (and placed in the *tab* folder), so they can be merged with other component-specific section files on a database build.
- The tool optionally extracts Dynamic Evidence links to Application, Product and Integrated Cases, configurable via the extractor input parameters. Note that the extractor does not extract Product or Integrated Case configuration information, only the links to them.
- The extractor implements three extraction strategies (the one to be used is determined by the input parameters):
  - *Extract all Dynamic Evidence Types*: All active Dynamic Evidence Types on the system are extracted.
  - *Extract a list of Dynamic Evidence Types*: This strategy allows users to specify a list of Dynamic Evidence Types to extract (using a list of Evidence Type logical names).
  - *Extract a set of Dynamic Evidence Types identified by Evidence Type code prefix*: Dynamic Evidence Type codes are generated using a customizable three character prefix e.g. *DET*. This extraction strategy allows users to only extract Dynamic Evidence Types that use a specific code prefix.
- Dynamic Evidence Types have localizable descriptions. Prior to Cúram V6.0 SP2 the descriptions for all Dynamic Evidence Types on the system were stored in a single properties file (*DynEvd\_EvidenceTypeDescriptions.properties*) in the *AppResource* entity. In Cúram V6.0 SP2 this mechanism was changed to store Dynamic Evidence Type descriptions in individual properties resources, one per Evidence Type. If the extractor finds the old evidence descriptions properties file, it will split it into individual Evidence Type specific properties files.



- The extractor extracts the key set (DYNEVDCODE) used to generate Dynamic Evidence Type codes (preserving the Next Unique Block ID for this key set). Dynamic Evidence uses the Cúram key server ability to generate human readable keys. This is used to generate Dynamic Evidence Type codetable codes. When the database is reset, the key generation mechanism is also reset and there is a likelihood of producing keys that clash with previously generated ones. This is best avoided by preserving the state of the Dynamic Evidence key set used to generate Evidence Type codetable codes.
- The extractor extracts both Active and In Edit Evidence Type versions.
- The extractor extracts snapshots from the `EVIDENCETYPEVERSIONSNAPSHOT` table.
- A number of rulesets related to Evidence Type Versions are extracted: generated data and processing rulesets, and custom (calculated attributes, validations, summary information) rulesets. Custom rulesets can be edited by customers. Newly created custom rulesets are extracted by the tool (together with In Edit Evidence Type Versions). However, any changes made to published custom rulesets will not be picked up by the extractor until the changes are published.
- The extractor does not extract localizable resource bundles referenced from rulesets. Rulesets can include localizable resource messages which are stored in resource bundles in the *AppResource* entity. If users choose to use such messages in customized rulesets they have to handle the extraction of containing resource bundles manually.
- The extractor can either be run from the command line or the Eclipse development environment.
- Before uploading extracted artefacts back to the database, codetables have to be re-generated (via the `ctgen` target or server build) to include the extracted Dynamic Evidence Type codetable codes.

### Running the Extractor

Before you run the Extractor, you might need to customize the application property

```
curam.dynamicEvidence.generated.business.object.tabs.sectionscuram.dynamicEvidence
```

The property can be found in the **Dynamic Evidence - configuration** section of the administration application. This property is used to specify a list of sections to which generated Dynamic Evidence tabs are added. The list must contain a comma separated list of section IDs. This list is used to generate section contribution (`.sec`) configuration files (written to the `tab` folder). The default section list is:

```
DefaultAppSection,SUPERAPPSection,AUDITORAPPSection,AUDITCOAPPSection,
INVESTAPPSection,FINAPPSection
```

The extractor can be run in two ways:

- Running the extractor from the command line:
  - Set `SERVER_DIR` and `CURAMSDEJ` variables if not already set. Depending on the project setup, `PRE_CLASSPATH` variable may have to be set to include any referenced libraries.
  - From the `EJBServer` folder call `./components/DynamicEvidence/script/extractDynamicEvidenceConfiguration.bat` file passing all input parameters using the `-D` option, for example:

```
./components/DynamicEvidence/script/
extractDynamicEvidenceConfiguration -
Dcomponent=MyComponent -DlowerKey=20000 -DupperKey=20999 -
DetExtractionList="etLogicalName1;etLogicalName2;etLogicalName3"
```

- Alternatively from the *EJBServer* folder call **components/DynamicEvidence/script/extractDynamicEvidenceConfiguration.xml** ant script using the -D option to pass input parameters, for example:

```
ant -f ./components/DynamicEvidence/script/
extractDynamicEvidenceConfiguration.xml -
Dcomponent=MyComponent - DlowerKey=20000 -DupperKey=20999 -
DextractAll=true
```

- Running the extractor from the Eclipse development environment:
  - In Eclipse create a new Java Application Run Configuration.
  - Specify the main class to be `curam.util.impl.BatchLauncher`.
  - Two program arguments must be specified (in the **Arguments** tab of the run configuration set up): the first argument is the batch operation to be run by the batch launcher, the second argument is the list of all input parameters to the extractor:

```
curam.dynamicEvidence.sl.util.configextractor.intf.
DynamicEvidenceConfiguration Extractor.extractConfigArtefacts

serverDir=D:\CC\DynamicProductWS
\DynamicProductMain\EJBServer,component=
MyComponent,lowerKey=20000,upperKey=20999,etCodePrefix=DET
```

The class path will have to be set to include all referenced libraries (depending on the project setup).

Note the *serverDir* parameter. It contains the location of the *EJBServer* folder. The list of input parameters is comma separated and should not contain any spaces.

### **Extractor input parameters**

When you run the Extractor from the command line, ensure that all input parameters are passed using the -D option, that is, **D<parameter name>=<parameter value>**.

Mandatory Parameters:

- component* - the name of the destination component folder where the extractor output is written (e.g. 'custom'). If this folder does not exist in `SERVER_DIR/components` the extractor will report an error.
- serverDir* - this parameter is automatically set (from the `SERVER_DIR` variable) when running the extractor from the command line. It must be specified if running the extractor from Eclipse, and defines the location of the 'EJBServer' folder.
- lowerKey* - specifies the end or the starting key for the key range used to generate replacement primary keys for extracted database records.
- upperKey* - specifies the starting or the end key for the key range used to generate replacement primary keys for extracted database records.

Parameters defining the Evidence Type extraction strategy, i.e. which evidence types to extract. At least one of these parameters has to be specified or the extractor will report an error:

- extractAll* - if this parameter is provided with a value of 'true', all Active Dynamic Evidence Types on the system (in the `EvidenceTypeDef` table) will be extracted. This parameter overrides the other extraction strategy parameters. If it is absent or its value is not 'true' the subsequent parameters will be considered.

- *etExtractionList* - specifies a list of Dynamic Evidence Types to be extracted. The list must contain one or more Evidence Type logical names separated by semicolons (no white space allowed). This parameter overrides the *etCodePrefix* parameter.
- *etCodePrefix* - specifies a Dynamic Evidence Type code prefix, e.g. 'DET'. Dynamic Evidence Types with codes beginning with the specified prefix will be extracted. The code prefix is a customizable application property (curam.dynamicEvidence.type.code.prefix in Dynamic Evidence – located in the configuration section of the administration application).

#### Optional Parameters:

- *extractProductLinks* - enables extraction of Dynamic Evidence Type links to Products. If set to true the relevant records from ProductEvidenceTypeDefLink entity will be extracted.
- *extractICLinks* - enables extraction of Dynamic Evidence Type links to Integrated Cases. If set to true the relevant records from AdminICEvidenceTypeDefLink entity will be extracted.
- *extractCaseLinks* - enables extraction of Dynamic Evidence Type links to Application Cases. If set to true the relevant records from CaseConfigEvidenceLink entity will be extracted.
- *datamanagerDir* - name of the directory where datamanager (DMX, blob, clob) files are to be written. This directory is located in the destination component directory and it is automatically created if it does not exist. The default directory name is 'data'.
- *dmxDir* - name of the directory where DMX files are written. This directory lives in the *datamanagerDir* directory and it is automatically created if it does not exist. The default directory name is 'initial'. Inside this folder, the tool creates directories for blob and clob files.
- *codetableDir* - name of the directory where codetable (CTX) files are written. This directory is located in the destination component folder and it is created automatically if it does not exist. The default directory name is 'codetable'.
- *tabDir* - name of the directory where section configuration (SEC) files are written. This directory is located in the destination component folder and it is created automatically if it does not exist. The default directory name is 'tab'.
- *preserveRangeKeys* - This parameter specifies a list of Range Keys. Primary keys of extracted database records will remain unchanged if the key falls inside one of the rangeKeys specified in this property. If not withing any range specified the primary key will be generated based on the `upperKey` and `lowerKey` properties specified. Specify as follows:- `-DpreserveRangeKeys="20000-20999,23000-23999"`
- *java.extra.vargs* - This parameter helps specify the arguments to the Java virtual machine. This helps resolve any occurrence of Out of memory runtime errors. Example : To set PermGen space while running the extractor, specify as follows:- `-Djava.extra.vargs="-XX:PermSize=128m"`

### Extracted artefacts

Codetables are extracted and saved in files with `ctx` extensions.

The files are written into a folder located in the destination component (see `codetableDir` input parameter). Each codetable is extracted into a separate file. Two codetables are extracted:

- EvidenceType
- TemporalEvTypeApproval

#### Entities (Database Tables):

Entities are extracted and saved in files with `dmx` extension. There is one such file per entity. These files are written to the data manager folder located in the destination component (see

*datamanagerDir* and *dmxDir* input parameters). Blob and clob records are extracted into external files and referenced from DMX files. Blob and clob files are saved into separate folders (named *blob* and *clob*) located inside the *dmxDir* folder.

- **EvidenceTypeDef**: this entity stores the Dynamic Evidence Types on the system and it is the first to be extracted. The Evidence Type extraction strategies mentioned earlier determine what records to extract from this entity. If no records can be extracted from **EvidenceTypeDef** the extraction process terminates.
- **EvidenceTypeVersionDef**: extraction of this entity is dependent on **EvidenceTypeDef**. It stores multiple metadata Versions for Dynamic Evidence Types. Active and In Edit Versions are extracted for each extracted record (Evidence Type) from **EvidenceTypeDef**.
- **EvidenceTypeDefinition**: maps an Evidence Type to an Evidence Nature. Extraction is dependent on **EvidenceTypeDef** entity.
- **SecurityGroup**: contains security group definitions on the system. There is an auto generated Security Group for each active Dynamic Evidence type, hence this entity is dependent on **EvidenceTypeDef** in the extraction process. Users have access to Security Groups via the System Administration application. If auto-generated Dynamic Evidence Security groups are modified or deleted they will not be picked up by the extractor. Additionally, Security Identifiers and Security Group SID mappings will not be extracted for modified groups.
- **SecurityIdentifier**: this entity depends on **SecurityGroup** and **SecurityGroupSid** in the extraction process. Only Security Identifiers linked to extracted security groups are extracted.
- **SecurityGroupSid**: links Security Identifiers to Security Groups. If any of the extracted Security Identifiers are linked to the EVIDENCEGROUP security group, these links are also extracted.
- **AppResource**: this entity stores miscellaneous information in blob fields. It depends on **EvidenceTypeDef** and **EvidenceTypeVersionDef** in the extraction process. The following artefacts are extracted:
  - Evidence Type localizable descriptions: extracted as properties files in the *blob* folder, one per Dynamic Evidence Type.
  - Evidence Type Version localizable properties: extracted as properties files in the *blob* folder, one per Active Evidence Type Version.
  - Tab configuration files: extracted as XML files in the *blob* folder. Up to three files are extracted per Active Evidence Type Version (tab, menu, navigation configuration files).
  - The previously used single evidence descriptions properties resource (*DynEvd\_EvidenceTypeDescriptions.properties*), if found, is split into individual (one per Evidence Type) properties files and stored in the blob folder.
- **CreoleRuleset**: this entity stores published rulesets and it is dependent on **EvidenceTypeDef** and **EvidenceTypeVersionDef** entities. Data and processing rulesets are determined from extracted **EvidenceTypeDef** records, while custom defined rulesets (calculated attributes, validation and summary information) are determined from extracted active **EvidenceTypeVersionDef** records.
- **CreoleRulesetEditAction**: stores In Edit rulesets and has dependency on **EvidenceTypeDef** and **EvidenceTypeVersionDef** entities. Data and processing rulesets are determined from extracted **EvidenceTypeDef** records, while custom defined rulesets are determined from extracted In Edit **EvidenceTypeVersionDef** records.
- **CreoleRulesetCategoryLink**: links rulesets to categories. It is dependent on both **CreoleRuleset** and **CreoleRulesetEditAction** in the extraction process

- `EvidenceRulesetDef`: this entity maps an Evidence Type to In Edit data and processing rulesets and to a Rule Object Propagator Configuration.
- `RuleObjectPropagatorConfig`: stores Propagator configurations, one for each Active Dynamic Evidence Type. `EvidenceRulesetDef` links this entity to Dynamic Evidence Types (`EvidenceTypeDef`).
- `LocalizableText`: this entity links Propagator configurations to their localizable descriptions. It has a dependency on `RuleObjectPropagatorConfig` for extraction.
- `TextTranslation`: stores Propagator configuration descriptions and depends on `LocalizableText` entity for extraction.
- `ProductEvidenceTypeDefLink`: this entity links Dynamic Evidence Types to Products and it is optionally extracted. It depends on `EvidenceTypeDef` entity in the extraction process.
- `AdminICEvidenceTypeDefLink`: this entity links Dynamic Evidence Types to Integrated Cases and it is optionally extracted. It depends on `EvidenceTypeDef` entity in the extraction process.
- `CaseConfigEvidenceLink`: this entity links Dynamic Evidence Types to Application Cases and it is optionally extracted. It depends on `EvidenceTypeDef` entity in the extraction process.
- `KeyServer`: only a single record is extracted from this entity. This is the key set (DYNEVDCODE) used to generate Dynamic Evidence Type codes.

#### Section Configuration Files:

Section configurations are stored in the `AppResource` entity. Each section configuration may refer to multiple components which makes extraction via DMX files impossible. Instead, section configurations are extracted into section contribution files (with extension `sec`) and saved in the `tab` folder (see `tabDir` input parameter).

## Features of the Dynamic Evidence Metadata Loader

Use the Dynamic Evidence Metadata Loader to import and export dynamic evidence configurable metadata.

A configuration goal of dynamic evidence is to provide an administrative-time tool to reduce the complexity and the expense in developing and maintaining the artefacts that Cúram Administrators need to define. Initially, the tool was targeted at customers who did not have development installations of Cúram. That is, after the customers created or maintained dynamic evidence type versions, the database became the system of record for the administered and generated artefacts. Database backups ensured that the data was persisted.

Over time, a dynamic evidence type can evolve. For example, a change in legislation might require that a new evidence attribute must now be recorded, starting from a specified date. Dynamic evidence supports this requirement by using dynamic evidence type versions to record modifications to metadata over time. As metadata evolves, it was necessary for customers with Cúram development projects to import and export the configurable metadata by using the Data Manager. This way, when the database was re-created, which is a typical development-time activity in Cúram, any dynamic evidence type versions are persisted and do not need to be re-created every time.

To help this effort, a Dynamic Evidence Metadata Loader tool is provided with Cúram. This tool includes the following two features:

- Download: The tool downloads dynamic evidence metadata information of an effective date from a runtime Cúram database and then writes metadata information in an XML file to the file system.
- Upload: The tool uploads dynamic evidence metadata information for an effective date from a file system and updates a record to a runtime Cúram database.

### ***Running the downloader***

You can run the downloader from the command line or from the Eclipse development environment.

The Downloader can be run in two ways:

- Running the Downloader from the command line:
  - From the EJBServer folder call `./components/DynamicEvidence/script/downloadDynamicEvidenceMetadata.bat` file passing all input parameters using the -D option, for example:
 

```
./components/DynamicEvidence/script/
downloadDynamicEvidenceMetadata -
Dlocation="DirectoryLocation" -DevidenceType="etLogicalName"
-DeffectiveDate="etvEffectiveDate"
```
  - Alternatively from the 'EJBServer' folder call `components/DynamicEvidence/script/downloadDynamicEvidenceMetadata.xml` ant script using the -D option to pass input parameters, for example:
 

```
ant -f ./components/DynamicEvidence/script/
downloadDynamicEvidenceMetadata.xml -
Dlocation="DirectoryLocation" -DevidenceType="etLogicalName"
-DeffectiveDate="etvEffectiveDate"
```
- Running the Downloader from the Eclipse development environment:
  - In Eclipse create a new Java Application Run Configuration.
  - Specify the main class to be `curam.util.impl.BatchLauncher`.
  - Two program arguments must be specified (in the 'Arguments' tab of the run configuration set up): the first argument is the batch operation to be run by the batch launcher, the second argument is the list of all input parameters to the extractor:

```
curam.dynamicEvidence.sl.util.metadataloader.intf.
DynamicEvidenceMetadataDownloader.downloadMetadata
location=DirectoryLocation,evidenceType=etLogicalName,
effectiveDate=etvEffectiveDate
```

The class path will have to be set to include all referenced libraries (depending on the project setup).

### ***Downloader input parameters***

When you run the downloader from the command line, ensure that all input parameters are passed using the -D option, that is, `-D<parameter name>=<parameter value>`.

Mandatory Parameters:



- *location* - the name of the destination folder where the Downloader output is written. If this folder does not exist or does not have privilege to create the folder in the file system the extractor will report an error.
- *evidenceType* - specifies the logical name of the Dynamic Evidence Type to be extracted.
- *effectiveDate* - specifies the effective date of a Dynamic Evidence Type Version to be extracted.

### **Running the uploader**

You can run the uploader from the command line or from the Eclipse development environment.

The Uploader can be run in two ways:

- Running the Uploader from the command line:
  - From the *EJBServer* folder call `./components/DynamicEvidence/script/uploadDynamicEvidenceMetadata.bat` file passing all input parameters using the `-D` option, for example:
 

```
./components/DynamicEvidence/script/
uploadDynamicEvidenceMetadata -Dlocation="XML
File Location" -DevidenceType="etLogicalName" -
DeffectiveDate="etvEffectiveDate"
```
  - Alternatively from the *EJBServer* folder call `components/DynamicEvidence/script/uploadDynamicEvidenceMetadata.xml` ant script using the `-D` option to pass input parameters, for example:
 

```
ant -f ./components/DynamicEvidence/script/
uploadDynamicEvidenceMetadata.xml -Dlocation="XML
File Location" -DevidenceType="etLogicalName" -
DeffectiveDate="etvEffectiveDate"
```
- Running the upload from the Eclipse development environment:
  - In Eclipse create a new Java Application Run Configuration.
  - Specify the main class to be `curam.util.impl.BatchLauncher`.
  - Two program arguments must be specified (in the **Arguments** tab of the run configuration set up): the first argument is the batch operation to be run by the batch launcher, the second argument is the list of all input parameters to the extractor:

```
curam.dynamicevidence.sl.util.metadataloader.
intf.DynamicEvidenceMetadataUploader.uploadMetadata
location=XML File Location,
evidenceType=etLogicalName,effectiveDate=etvEffectiveDate
```

The class path will have to be set to include all referenced libraries (depending on the project setup).

### **Uploader input parameters**

When you run the uploader from the command line, ensure that all input parameters are passed using the `-D` option, that is, `-D<parameter name>=<parameter value>`

Mandatory Parameters:

- *location* - the location of the XML file that contains metadata information in the file system. If this file does not exist or does not have valid contents the Uploader will report an error.

- *evidenceType* - specifies the logical name of the Dynamic Evidence Type to be uploaded.
- *effectiveDate* - specifies the effective date of a Dynamic Evidence Type Version to be uploaded.

## Generated artefacts

Dynamic evidence generates a number of artefacts automatically as dynamic evidence types are administered. Use this information to understand what these artefacts are, when they are generated, and which actions you can take on them.

Administrators must understand generated artefacts because the artefacts appear in the Cúram administration screens. To remove or alter artefacts can result in runtime problems with the operation of Dynamic Evidence Types.

Administrators must also understand that by deleting a Dynamic Evidence Type removes the generated artefacts and can affect code or configurations that depend on generated Dynamic Evidence artefacts.

### ***EvidenceType codetable entries***

When a new dynamic evidence type is created, a code table entry is automatically added to the `EvidenceType` code table.

The generated Codetable code will begin with the value of the `curam.dynamicEvidence.type.code.prefix` environment variable, and then a generated unique number will be appended. The Codetable item description will be the name of the new Dynamic Evidence Type as entered by the administrator on the Dynamic Evidence Type Create page. The locale will be the server locale.

The Codetable code prefix is limited to three characters and its default value is *DET*.

Evidence Type Codetable entries will be removed if the Dynamic Evidence Type is cancelled.

**warning** Auto-generated Codetable entries are recorded against the server locale, but displayed for the locale of the currently logged in user. It is highly recommended that Administrator users configuring Dynamic Evidence operate in the same locale as the server. In multi locale deployments this will prevent localization issues that may occur before auto-generated Codetable items are translated for all supported locales.

### ***Property resources for evidence types***

Each dynamic evidence type has a property application resource that contains the localizable evidence type description property.

This resource is created when the Evidence Type is created. The property resource name is made up of “`DynEvd_EvidenceType_`” followed by the logical name of the Dynamic Evidence Type and ended with “.properties” extension (e.g. `DynEvd_EvidenceType_<logicalName>.properties`).



### ***Security identifiers and security groups***

Each dynamic evidence type is associated with a security group. The name of the security group is specified by the administrator when the dynamic evidence type is created. The security group is automatically added.

Security Identifiers (SIDs) are created for evidence record maintenance operations when Dynamic Evidence Types are created. Three SIDs are generated for each Dynamic Evidence Type: one each for create, modify and read of evidence record in respect of the Dynamic Evidence Type.

These are named “DynEvd.create.*EvidenceTypeGroupName*”, “DynEvd.modify.*EvidenceTypeGroupName*”, and “DynEvd.read.*EvidenceTypeGroupName*”, where *EvidenceTypeGroupName* should be replaced with the name of the Security Group the administrator chose for the Dynamic Evidence Type.

The generated SIDs are automatically added to both the Security Group specified by the administrator, and to the EVIDENCEGROUP Security Group, if it exists.

Generated SIDs and Security Groups are removed if a Dynamic Evidence Type is cancelled.

### ***Cúram Express Rule (CER) Rule Sets***

When each new dynamic evidence type version is activated, several Cúram Express Rule (CER) rule sets are generated. The generated rule sets are removed if the dynamic evidence type version is cancelled.

Administrators should refer to [Dynamic evidence rule sets on page 77](#) for more information on generated Rule Sets.

### ***Propagator configuration***

Propagator configurations are generated corresponding to the generated data rule sets.

Administrators should see [Dynamic evidence rule sets on page 77](#) for more information on dealing with generated Propagator Configurations.

### ***Properties resources for evidence type versions***

A property application resource is generated on activation for each dynamic evidence type version that contains all localizable strings that are associated with its user interface.

The naming convention for these is “DynEvd\_” followed by the logical name of the Dynamic Evidence Type and then the Effective Date of the Dynamic Evidence Type Version. This Property Application Resource is generated in the default locale and is created on activation of the Dynamic Evidence Type Version. Administrators should not modify or remove these resources.

These resources are removed when a Dynamic Evidence Type Version is cancelled.

For more information about these resources, please see [Localizing dynamic evidence on page 93](#).

### ***Dynamic UIM resources***

Each dynamic evidence type version has a number of dynamic UIM pages and corresponding property application resources that are associated with it.

These are generated on the first page access in the case worker application. All such files are prefixed with a “DynEvd\_” prefix. Administrators should not edit or remove these Resources.

These Resources are removed when a Dynamic Evidence Type Version is cancelled.

### ***Tab configuration and the Application Resource Store***

The dynamic evidence runtime user interface requires some tab configuration resources. These are stored in the Application Resource Store.

Each Dynamic Evidence Type Version will have a corresponding tab resource. The Resource name will be the Dynamic Evidence Type code concatenated with the Effective Date of the Dynamic Evidence Type Version. There will also be a nav configuration Resource, the name of which will be identical to that of the tab configuration Resource, but with the word “Nav” appended.

If the Dynamic Evidence Type is parent to another Dynamic Evidence Type, then a menu configuration Resource will also be generated. The name of this Resource will be identical to that of the tab resource with the word “Menu” appended.

All Section resource files for each Cúram Application View will also be updated to include these generated tab configurations when a new Dynamic Evidence Type Version is activated.

If a Dynamic Evidence Type Version is cancelled, the tab configuration Resources associated with it will be removed, and all references in the Application section files will be cleaned up.

### ***Domain definitions***

Dynamic evidence generates domain definitions dynamically where required.

These are stored in the Application Resource store in a Resource called *DynEvdDomains.xml*. This is a single Resource that is shared between all Dynamic Evidence Types. It will be generated on startup when the client application attempts to load it, and will be regenerated each time a new Dynamic Evidence Type Version is Activated or Cancelled. Administrators should not modify or remove this Resource.

### ***The Evidence Type Definition entity***

The `EvidenceTypeDefinition` entity is used to store information about the behavior of an evidence type. For each new dynamic evidence type an entry is added to this database table.

### ***Summary of generated artefacts***

Use a generated artefacts summary as a checklist.

This section is a summary of generated artefacts. It is intended as a useful checklist to accompany the preceding information.

*Table 38: Summary of Generated Artefacts*

<b>Name</b>	<b>Type of artefact</b>	<b>Multiplicity</b>
curam.dynamicEvidence.type.code.entity plus a unique identifier	Entity in EvidenceType codetable	One codetable entry per evidence type
“DynEvd_EvidenceType_” <i>ET</i> Logical Name “.properties”	Properties resource in AppResource store	One per Dynamic Evidence Type
Specified by administrator on evidence type creation	Security Identifiers (SIDs)	Three SIDs per evidence type - create, modify and read
<i>Evidence Type Logical Name</i> and various suffixes	CER Rulesets	Multiple rulesets per evidence type
Corresponding to the ruleset names	Ruleset propagator configurations	One propagator configuration per generated ruleset

Name	Type of artefact	Multiplicity
"DynEvd_" <i>ET Logical Name</i> plus a number	Properties resource in AppResource store	One per evidence type version
"DynEvd_" and various suffixes	Dynamic UIM and corresponding properties resource in AppResource store	Several pairs per evidence type version
Dynamic Evidence Type code concatenated plus a number	Tab Configuration in AppResource store	One per evidence type version
The same as the tab configuration resource name but with "Nav" appended	Nav Configuration in AppResource store	One corresponding to each tab resource
The same as the tab configuration resource name but with "Menu" appended	Menu Configuration in AppResource store	One for each evidence type that has child evidence types
Named for the Cúram application view	Section resource files in AppResource store	Each Cúram application view has a "Section" resource
DynEvdDomains.xml	Entry AppResource store	One
EvidenceTypeDefinition	Database table	One entry on the table per evidence type

## Dynamic Evidence application properties

Cúram application properties are administered through the Cúram system administration application.

For more information, see the *System Administration Guide*.

Table 39: Dynamic Evidence Application Properties

Property name	Description	Default
curam.dynamicevidence.type.codetable.prefix	Defines the auto-generated Dynamic Evidence Type Codetable codes. Prefix length is limited to three characters.	DET
curam.dynamicevidence.inprogress.evidence.size	Dynamic Evidence contains a LRU cache that is used to temporarily store data for in-progress user operations where the user transfers between interfaces for different versions of the same evidence type. This property specifies the maximum size for this cache.	1000

## Compliance for dynamic evidence

Specific information about dynamic evidence compliancy and upgrades.

**Java API**

Dynamic evidence provides public APIs that might be invoked from custom rule sets and application code.

Nothing will be changed or removed from this public API without following standards for handling customer impact. These APIs can be identified by looking at the Javadoc information shipped with Dynamic Evidence.

Unless explicitly permitted in the Javadoc information, you must not provide your own implementation of any Dynamic Evidence Java interface, nor subclass any Dynamic Evidence implementation Java class.

**Infrastructure rule sets**

Dynamic Evidence ships with four infrastructure rule sets.

Dynamic evidence ships with the following infrastructure rule sets:

- `DynamicEvidenceRuleSet`
- `EvidenceCalculatedAttributesRuleSet`
- `EvidenceSummaryRuleSet`
- `EvidenceValidationRuleSet`

These Rule Sets should not be modified as part of custom development as these act as the contract between Dynamic Evidence and custom Dynamic Evidence Processing Rule Sets.

**Dynamic evidence types prefix**

A codetable entry is generated in the `EvidenceType` codetable for each dynamic evidence type. By default, the generated entries include a 'DET' prefix.

As per the general compliancy rules regarding Codetable entries, customers should not use the "DET" prefix for their Dynamic Evidence Types. Administrators should use the application property `curam.dynamicEvidence.type.code.prefix` to define a different prefix for their custom Dynamic Evidence Types. Refer to [Dynamic Evidence application properties on page 107](#) for more information on this property.

**Source code for the sample widgets**

Complete three steps for the sample widget.

- **Step 1**

Configure the domain name and its plug-in in the `DomainsConfig.xml` file in `\webclient\components\<component-name>`

```
<dc:domain name="EXAMPLE_DOMAIN_NAME">
  <dc:plug-in name="edit-renderer" class="EditRenderer"/>
  <dc:plug-in name="view-renderer" class="ViewRenderer"/>
  <dc:plug-in name="converter" class="DomainConverterPlugin"/>
</dc:domain>
```

- **Step 2**

Implement the class plug-in in `\webclient\components\<component-name>\javasource`

For examples, see the *Custom Widget Development Guide*.

- **Step 3**

Once the implementation is done, build the client for the changes to be reflected.

Now "EXAMPLE\_DOMAIN\_NAME" can be used as domain name in dynamic evidence configuration

For more information about developing Domain and its custom widget, see the *Custom Widget Development Guide*.



# 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.