IBM Cúram Social Program Management
Version 7.0.0

*Cúram Evidence Generator
Specification*
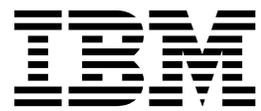
IBM

Edition

This edition applies to IBM Cúram Social Program Management v7.0.0 and to all subsequent releases unless otherwise indicated in new editions.

Licensed Materials - Property of IBM.

Contents

Figures

Tables

Evidence Generator Specification

Use the Cúram Evidence Generator to rapidly develop server-side code and client side screens for evidence. Various design patterns can be applied during development. Certain types of metadata are required for each evidence type. Metadata controls the generated evidence output.

Introduction

Purpose

The purpose of this document is to provide a complete reference for the configuration of the Cúram Evidence Generator, as well as for the development of evidence entities that make use of it.

It will describe in full, all the patterns that can be applied at development time, the meta data required for each, and the effect it will have on the generated output

Prerequisites

The reader should be familiar with the evidence capturing aspect of case management as well as its use in determining eligibility and entitlement on a case. They should also have read the Evidence High Level Architecture as outlined in Chapter 1 of the *Designing Cúram Evidence Solutions* guide. Although a substantial amount of code is generated, it would be useful for the reader to familiarise themselves with the *Cúram Evidence Developers Guide*.

Audience

This document is targeted at a technical audience, both developers and architects, intending to generate evidence solutions using Cúram's Evidence Generator.

Overview

Introduction

The Cúram Evidence Generator is intended as a rapid way of developing the server side code, and client side screens for evidence entities, that integrates fully with the standard Cúram Evidence Solution.

Note: The generator requires that the entity is modeled with specific options set, and that certain associated structs are created according to a naming convention and with specific aggregations. Full details of this can be found in the *Cúram Evidence Generator Modeling Guide* and as such is not covered here.

Inputs

The generator relies on a number of input sources in order to be able to produce its outputs.

Configuration
The generator is configured via a components `evidence.properties` file. This provides options for, amongst other things, setting the code package of generated code, the location of generated files etc.

Standard Properties and Message Files

Each component must also provide a general message, and two standard properties files:

general.properties: This file provides many of the standard properties needed by the generator, such as various page and list titles and standard action control links. It helps to ensure a consistent experience across all evidence types under the generators control.

employment.properties: This file provides many of the standard properties needed by the generator when linking with core Employments, such as various page and list titles and standard action control links. It helps to ensure a consistent experience across all evidence types under the generators control.

Ent<product.prefix>GeneralError.xml: This file provides many of the error messages that the generated code will attempt to throw under certain circumstances.

Entity Meta Data

Each entity being generated requires its own Meta Data files to be provided. These describe various features of the entity being generated, and are roughly separated into two distinct sections.

Server Side: The server side meta data is used to define various things such as the relationships between various entities, which participant the record will relate to, the business dates of the entity and any cached database operations.

Client Side: The client side meta data is more concerned with the layout of the generated screens, and the text labels and descriptions that appear on them.

Outputs

The Cúram Evidence Generator produces all the code, screens, and configuration files required for evidence types to fully integrate with the standard Cúram Evidence Solution, without any further coding required.

The generator also produces skeleton implementations of various extension points in the code, in order to allow simple customisation of the generated evidence for things such as validations, and both pre and post processing for the standard Create, Read and Update methods.

Modeling

For each entity being handled by the generator, a service layer class as well as various extension classes are modelled. This ensures that all generated code can be accessed via the normal application interface / factory method.

In addition, a facade class is generated per component to provide access to all the CRUD operations for those entities.

Implementation Code

Implementation code is generated for all of the modeled classes created by the generator, as well as for the entity layer. This ensures that there is no requirement to write any further code.

There are however several extension points generated where custom code can be easily integrated into the generated implementation code. This is useful for things such as validations, evidence object descriptions etc.

Message Files

The generator also produces a message file per entity with specific error messages contained within.

Screens

The generator writes all the screens required for creating, modifying, viewing, listing etc the different evidence records, as well as any resolve scripts required to integrate the generated screens with the standard infrastructure screens.

Wizards: When choosing to create an evidence object at runtime, often related / parent objects must be chosen. In this case the generator will produce all the wizard configuration and screens required to take the user through this process, step by step.

Tabs: Each entity will also have a Business Object Tab produced to allow the user view all details about a given evidence object, such as its change history, and any related objects.

Setup & Configuration

Introduction

The purpose of this chapter is to explain how to setup your stream to use the generator, and also look at the configuration options provided.

Directory Structure

The Cúram Evidence Generator is designed to automatically find any locations where evidence needs to be generated by looking for a specific directory and file structure within each component in the component order. As such it is very important to get the structure correct.

Note: The case of letters in the directory and file names is important and should be created exactly as specified.

Base Directory

The base directory for the configuration and meta data should be named `evidence`, and the file `evidence.properties` should exist.

Properties Directory: Within the `evidence` directory there should be a directory called `properties`. This is the location for the `general.properties` and `employment.properties` files.

Server Meta Data Directory: Within the `evidence` directory there should be a directory called `server` and within this a directory called `metadata`. This will be the location for your server meta data files.

Client Meta Data Directory:

Integrated Case Level

Within the `evidence` directory there should be a directory called `integrated` and within this a directory called `metadata`. This will be the location for your client meta data files for integrated level cases.

Product Delivery Case Level

Within the `evidence` directory there should be a directory called `product` and within this a directory called `metadata`. This will be the location for your client meta data files for product delivery level cases.

Configuration

The `evidence.properties` file allows customisation of different aspects of the Cúram Evidence Generator, ranging from the location of generated output files, to the java code package used. There are a lot of options, which are dealt with in full in the appendix "evidence.properties" on page 12.

Common Page Properties

general.properties

The `general.properties` file is used to specify common properties that are used on many generated pages. This includes everything from page titles and list column headers, to labels for common actions. Again there are a lot of options here, which are dealt with in full in the appendix "general properties" on page 14.

employment.properties

The `employment.properties` file is used to specify common properties that are used on generated pages involving relating generated evidence to the core Employment entity. There are a lot of options, which are dealt with in full in the appendix "employment.properties" on page 23.

Server Meta Data

Introduction

The purpose of this chapter is to introduce the server side meta-data files and their contents, and discuss the different patterns that can be employed.

Generally speaking, the server side meta data is used to describe the relationships between entities, as well as several options regarding cached methods and the participant to associate the evidence to.

For a full description of the meta-data format, and possible values please see the appendix "Server Meta Data" on page 27.

The `Entity` Node

The root node of a server metadata XML document is the `Entity` node. This contains attributes for specifying the logical name of the Entity and an attribute to specify which case participant a record should be associated with.

```
<Entity logicalName="PaidEmployment"
        relateEvidenceParticipantID="employeeCPRID"
    >
    ...
    </Entity>
```

Relationships

The `Relationships` node is used for specifying information about the how the current entity relates to other evidence entities, as well as certain core entities. There are no attributes on this node, but the following sub patterns can be used:

Parent / Child Relationships

The 'Parent-Child' pattern describes a hierarchical relationship between two evidence entities. It is the fundamental relationship in Cúram Evidence. The relationship essentially means that the child cannot be created until a parent record has been created. This is enforced by the navigation within the application. The pattern can be used to create multi-tier (generational) relationships, e.g. Parent-Child-Grandchild-Greatgrandchild.

Metadata entries: The metadata for describing a Parent-Child relationship simply requires listing the possible Parent or Child types for each entity. In the following example, a Parent-Child relationship exists between Paid Employment (Parent) & Employment Address (Child).

From PaidEmployment.xml (i.e. the XML to describe the Paid Employment entity):

```
<Relationships>
        <Child name="EmploymentAddress" />
      </Relationships>
```

From EmploymentAddress.xml (i.e. the XML to describe the Employment Address entity):

```
<Relationships>
        <Parent name="PaidEmployment"/>
      </Relationships>
```

Multiple Mandatory Parents

The 'Multiple Mandatory Parents' pattern is similar to the standard Parent-Child pattern except that more than one parent must be specified for each.

Metadata entries: To describe a Multiple Mandatory Parents relationship, the list of Parent types should be wrapped in a `MandatoryParents` node.

Expanding on the example from the Parent / Child section, from EmploymentAddress.xml:

```
<Relationships>
        <MandatoryParents>
          <Parent name="PaidEmployment"/>
          <Parent name="AnotherParentType"/>
        </MandatoryParents>
      </Relationships>
```

Pre Association Relationships

The 'Pre-Association' pattern is used when an association exists between two entities and the user is required to select the associated record prior to creating the record that completes the association.

Metadata entries: The metadata for this pattern is very simple and involves simply specifying the list of entity types to be chosen as a pre association.

```
<Relationships>
        <PreAssociation to="AnotherEntityType"/>
      </Relationships>
```

Related Relationships

The 'Related' pattern is used to relate an evidence record to a non-evidence record. This is typically achieved by storing the unique identifier of the non-evidence record as a foreign key on the evidence entity. An example of this might be to

relate a Paid / Self Employment evidence record to a core Employment record. This is done by storing the unique identifier of the core Employment record on the Paid / Self employment entity.

A feature of the Related pattern is that it is necessary to specify a list of case participant roles in order to be able to list the related records that may be chosen.

Metadata Entries: Again, taking the example of a PaidEmployment evidence type, the following metadata would be used to allow the user to choose from a list of core Employments relating to case participants of types PRIMARY or MEMBER

```
<Relationships>
        <Related to="Employment">
          <ParticipantType type="PRIMARY"/>
          <ParticipantType type="MEMBER"/>
        </Related>
      </Relationships>
```

Business Dates

The Business Dates pattern is used to specify which, if any, of the date fields on the entity correspond to the business start and end dates of that entity. These dates are then used in the calculation of the period for which the evidence object applies.

The generator will return these dates from the getStartDate() and getEndDate() methods defined on the EvidenceInterface.

Metadata entries

Using the BusinessDates node, either the startDate, endDate or both can be specified as follows:

```
<Entity logicalName="PaidEmployment" ... >
        <BusinessDates startDate="employmentStartDate"
            endDate="terminationDate"/>
      </Entity>
```

Override

The Override node allows the developer to override an out of the box entity provided with Solution modules or to add a new custom entity to an out of the box product.

A full example of how to use this feature is provided in the Cúram Evidence Generator Cookbook

Cached Operations

The Cached Operation pattern allows the developer to cache SQL operations in the generated entity layer class. The generator uses a caching pattern to implement and manage the cache. The operation must be a SQL operation on the entity. This is because the entity class is generated so it is not possible to implement business logic within it.

Metadata entries

Looking again at the PaidEmployment example, to cache the readDetails method, the following code would be used:

```
<Entity logicalName="PaidEmployment" ... >
        <CachedOperation>
          databaseRead = "read"
          operationName = "readDetails"
          returnType =
```

```
                         "curam.example.evidence.entity.struct.
                             PaidEmploymentDtls"
                    </CachedOperation>
                  </Entity>
```

Client Meta Data

Introduction

The purpose of this chapter is to introduce the client side meta-data files and their contents, and discuss the different patterns that can be employed.

Generally speaking, the client side meta data is used to describe the layout of the screens that should be generated, as well as specifying how to choose values for case participant fields, and when to create new role types for those participants.

For a full description of the meta-data format, and possible values please see the appendix "Client Meta Data" on page 31.

The `Entity` Node

The root node of a client metadata EUIM document is the `Entity` node. This contains attributes for specifying the name of the Entity as well as the display name for use on screens. You can also specify here whether or not the entity should be modifiable.

```
<Entity name="PaidEmployment"
        displayName="Paid Employment"
        modify="Yes"
     >
     ...
     </Entity>
```

User Interface

The `UserInterface` node is the containing node for all UI elements. It has one attribute that allows you to specify whether or not the create screen for the entity should have a `Save & New` button.

```
<UserInterface saveAndNewButton="Yes">
       ...
       </UserInterface>
```

Clusters

As with developing in UIM, the EUIM `Cluster` Node is used to group UI elements together. There are a number of attributes available for specifying the number of columns, the label and description text. There are also 3 special attributes, create, modify and view, that allow the `Cluster` to be hidden, or shown on different screens, allowing a different layout to be used on create screens versus modify ones.

The following code creates one `Cluster` that is visible on create and modify pages only, and has a label specified by the property `Cluster.Label` in the associated properties file, as well as a second `Cluster` that is only shown on view pages.

```
<Clusters>
       <Cluster label="Cluster.Label" create="Yes" view="No">
       ...
       </Cluster>
```

```
        <Cluster label="Cluster2.Label" create="No" modify="No">
        ...
        </Cluster>
      </Clusters>
```

Fields

The `Field` node is used to specify an individual field on the screen. There are many attributes that can be used to control the behaviour of this node, from the database columnName its associated with, to the usage of blank or default values in codetable fields.

An additional attribute called `metatype` can also be used to control the behaviour of the field. Information on the available metatypes, and their effects can be found in the "Metatypes" on page 37 reference in the appendix.

Using an example of an entity attribute called `employmentType`, that is a codetable of possible employment types. The following code would produce a field on screen that started with the default value from the codetable, and once set on the entities create page, would not be modifiable from the entities modify page.

```
<Cluster ... >
      <Field columnName="employmentType" use_default="true"
        label="Field.EmploymentType.Label" modify="No"
      />
      </Cluster>
```

Case Participant Fields

When using a `Field` node to represent a case participant attribute on the entity, there are a number of further metadata entries required, that will be explained in this section.

Note: The `metatype` attribute of the `Field` node should be set to `CASE_PARTICIPANT_SEARCH`

Overview

There are three ways of specifying a participant on screens created by the generator.

- **Choosing from a drop down list**

  The metadata can be used to specify what case participant types should be included in the list

- **Searching**

  The system can be searched for an appropriate participant

- **Registering a new representation**

  A new representative can be added to the system.

CaseParticipant Node

The `CaseParticipant` Node provides extra information about the field and how the information is stored. It includes options for, amongst other things, telling the system to create a new case participant role for the chosen participant.

SearchType Node

This node is used to specify a particular type of case participant role that should be listed in the drop down select box.

`CreateCaseParticipant` Node

When choosing a pre existing case participant from the drop down list, there is the option of creating a new role for that participant, if they do not already have it.

For example, you could choose to populate the drop down with all case participants of type PRIMARY, MEMBER and ALIEN, and choose to create a new role for the participant of type ALIEN. Then, if the participant chosen was already of type ALIEN, no new role would be created, but if the type was either PRIMARY or MEMBER, it would.

Example

The following EUIM code, will provide a Case Participant field on screen where the drop down is populated with participants of type PRIMARY and MEMBER. As the create option is turned off, the end user will have no option to search the system for a participant, or to register a new representative.

```
<Cluster ... >
        <Field columnName="myCaseParticipant"
          label="Field.MyCaseParticipant.Label"
          metatype="CASE_PARTICIPANT_SEARCH"
        >
          <CaseParticipant create="No">
            <SearchType type="PRIMARY"/>
            <SearchType type="MEMBER"/>
          </CaseParticipant>
        <Field>
      </Cluster>
```

The following EUIM code, builds on the first example, and will now create a role type of MEMBER if the chosen participant does not already have that role. As well as the drop down list, this will allow the user to search the system for a suitable participant, as well as specifying a new representative.

```
<Cluster ... >
        <Field columnName="myCaseParticipant"
          label="Field.MyCaseParticipant.Label"
          metatype="CASE_PARTICIPANT_SEARCH"
        >
          <CaseParticipant create="Yes">
            <CreateCaseParticipant
              participantType="Person"
              roleType="MEMBER"
            />
            <SearchType type="PRIMARY"/>
            <SearchType type="MEMBER"/>
          </CaseParticipant>
        <Field>
      </Cluster>
```

Adding Functionality

Introduction

The purpose of this chapter is to explain how to extend the functionality of a generated entity, from adding validations, populating screen fields from sources other than the entity itself, providing a description for use on evidence workspace lists etc.

There are a number of extension classes that can be coded, with the generator providing a default skeleton implementation for each in your source code

directory. Additionally each of these classes is automatically modeled by the generator, so all follow the standard factory, interface, implementation pattern used in the application.

Customise Class

This class provides methods that get called at specific points within the generated service layer code, which allows developers to implement their own custom logic, as well as modifying the values being passed to and from the screen.

Class Name

The class is named `Customise<Entity Name>.java`

Package Name

The class is placed in the package

`curam.<product.package>.evidence.customise.impl`

where `product.package` is as specified in the `evidence.properties` file

Methods

The methods provided are as follows

Table 1. Customise Methods

Method	Details
preCreate	Allows custom processing to be performed before the evidence insert operation happens.
postCreate	Allows custom processing to be performed after the evidence insert operation happens.
preModify	Allows custom processing to be performed before the evidence modify operation happens.
postModify	Allows custom processing to be performed after the evidence modify operation happens.
preRead	Allows custom processing to be performed before the evidence read operation happens.
postRead	Allows custom processing to be performed after the evidence read operation happens.

Hook Class

The hook class provides the developer with access to a number of infrastructure methods from the `EvidenceInterface` that should be implemented for each entity. Normally the developer would implement these methods on the entities implementation class, but as this is now generated, the hook class must be used instead.

Class Name

The class is named `<Entity Name>Hook.java`

Package Name

The class is placed in the package

`curam.<product.package>.evidence.hook.impl`

where `product.package` is as specified in the `evidence.properties` file

Methods

The methods provided are as follows

Table 2. Hook Methods

Method	Details
calcAttributionDatesForCase	Returns the attribution dates for an entity.
getDetailsForListDisplay	Returns the textual description of an evidence object.

Validate Class

The validate class provides methods where custom validations can be added for an entity.

Class Name

The class is named `Validate<Entity Name>.java`

Package Name

The class is placed in the package

`curam.<product.package>.evidence.validation.impl`

where `product.package` is as specified in the `evidence.properties` file

Methods

The methods provided are as follows

Table 3. Validation Methods

Method	Details
preModifyValidate	Called from within the entities `preModify` method.
preInsertValidate	Called from within the entities `preInsert` method.
validate	Called from within the entities standard `validate` method.

Related Attributes Class

If using the related entity attributes pattern, a further class is generated that provides a method for reading these related values and returning them. This method is called during the service layers read operation, and also when the create screen is being loaded, so that the values can be displayed there as well.

Class Name

The class is named `<Entity Name>RelatedEntityAttributes.java`

Package Name

The class is placed in the package

`curam.<product.package>.evidence.relatedattribute.impl`

where `product.package` is as specified in the `evidence.properties` file

Methods

The methods provided are as follows

Table 4. Related Attribute Methods

Method	Details
getRelatedEntityAttributes	Method to read any related attributes from alternative sources.

evidence.properties

The `evidence.properties` file is used to configure the generator options.

Properties Explained

The following section provides a list of all the properties that can be set and what they are for

Table 5. Properties Explained

Property Key	Description
product.build.option	Defaults to `false`. Must be set to `true` to build or clean evidence for this product. Otherwise, the Evidence Generator will ignore evidence for this product.
product.name	Insert the product name here. It is used to specify where all generated files are output to.
product.ejb.package	Code package name used for all impl layer directories in the product for which evidence is being generated.
product.prefix	The prefix is prepended to the name of all generated UIM pages and certain generated classes, such as the façade
product.appendAltID	Flag to determine whether the primary alternate ID is appended to all Case Participant names (on generated evidence screens). Defaults to `false`.
product.component.root	Root directory specifying where generated server files are copied to.
product.evidence.build.root	Root directory specifying where all temp generated output is copied.
evidence.properties.dir	Location for individual products properties file (which contains all product building config information). This must end with an 'evidence' directory.
properties.home	Product properties directory. Contains properties files and localized values for product wide client screen label values.
product.webclient	Root directory specifying where generated client pages are output to.
server.evidence	Location of EvidenceEntities.xml output used for server-side and infrastructure generation.
casetype.product.evidence	Name and location of EvidenceEntities.xml output for caseType 'Product' used for client-side Product Delivery evidence screen generation.
casetype.integratedCase.evidence	Name and location of EvidenceEntities.xml output for caseType 'Integrated Case' used for client-side Integrated Case evidence screen generation.

Table 5. Properties Explained (continued)

Property Key	Description
server.metadata	Location of server XML files which describe the entity's relationships, function creation as well as infrastructure generation.
caseType.integratedCase.metadata	Location of EUIM files used for EvidenceEntities.xml generation used for client-side Integrated Case evidence screen generation.
caseType.product.metadata	Location of EUIM files used for EvidenceEntities.xml generation used for client-side Product Delivery evidence screen generation.

Sample File

There are four properties that should be set to specific values for your product, these are listed first in this sample file. Recommended values are provided for the others as they mostly relate to intermediary files produced during generation and so customisation offers no real advantage in most situations.

Note: There should be no line breaks in individual properties.

```
## Values Specific to your component
product.name = <Component Name>
product.prefix = <Chosen Prefix>
product.ejb.package = <Chosen Package>
product.webclient = ${sysenv.CLIENT_DIR}/components/<Component Name>

## Recommended Values
product.build.option = true
product.appendAltID = false
product.component.root =
   ${product.components.root}/${product.name}
evidence.properties.dir =
   ${product.components.root}/${product.name}/evidence
properties.home = ${evidence.properties.dir}/properties/
product.evidence.build.root =
  ${evidence.build.root}/${product.name}
server.evidence = ${product.evidence.build.root}/model/server
casetype.product.evidence =
   ${product.evidence.build.root}/model/product/
      EvidenceEntities.xml
casetype.integratedCase.evidence =
   ${product.evidence.build.root}/model/integrated/
      EvidenceEntities.xml
server.metadata =
   ${product.components.root}/${product.name}/evidence/
      server/metadata
caseType.integratedCase.metadata =
   ${product.components.root}/${product.name}/evidence/
      integrated/metadata
caseType.product.metadata =
   ${product.components.root}/${product.name}/evidence/
      product/metadata
create.employment.link = true
create.clientlist.for.employment  = false
```

general properties

The general.properties file contains all generic label values for the product. These consist of localized label values for all common buttons, page titles etc. Some also have the ability to allow for dynamic values, i.e. the name of the evidence entity the page title is describing. All properties within this file must be set.

Note: The property keys cannot be changed, added or removed. To do so would cause errors in the running of the evidence generator.

Dynamic Properties

All properties listed in this section should have some dynamic placeholder to give more meaning to their operation. The following section explains the use of dynamic placeholders and how to use them.

Dynamic Property Values

Dynamic Properties are those where there is an option to add a dynamic value to a property at generation time. This can be useful for page titles, menu options etc, or anywhere that further context is useful

This is achieved by using a placeholder where you wish the dynamic value to be placed during the generation of the properties file. There is currently one placeholder type supported by the evidence generator as can be seen listed below. This is directly related to metadata tags within the EUIM files.

Note: A dynamic placeholder can be used multiple times in a property value and or a combination of different placeholders. However, the user must be aware of the relationship between these placeholders and the actual evidence metadata that the evidence generator will process. A value will only be substituted into the placeholder if the metatype tag that this placeholder maps to exists in the Evidence Entity metadata.

<displayName>
> The name of the Evidence entity as it would appear on screen. This is not the same as the physical name that would appear on the table in the database, e.g.
>
> physical name = PaidEmployment
>
> display name = Paid Employment

Examples of Dynamic value usage: The following uses displayName as an example:

Using for this example the Page.Title.EntityWorkspace "Page Title Keys" on page 15 the value for this property would be entered as follows:
```
Page.Title.EntityWorkspace=<displayName> Evidence
```

At build time, the correct substitutions will be made when the evidence generator processes the EUIM files. Using the Paid Employment evidence entity for instance, the following property would be generated into the appropriate.properties files.
```
Page.Title=Paid Employment Evidence
```

Page Title Keys

This table describes the property keys for generic Page Title properties.

Table 6. Page Title Keys.

This table describes the property keys for generated Page Title properties.

Property Key	Description
Page.Title.EntityWorkspace	Title for the main page of the evidence workspace generated for each evidence type which is used in the evidenceFlow control.
Page.Title.ModifyEntity	Title used for the generated modify pages.
Page.Title.NewEntity	Title used for generated create pages.
Page.Title.ViewEntity	Title used for generated view pages.

Help.PageDescription Keys

UIM Pages use a property called `Help.PageDescription` to provide help for the page. This table describes the property keys that should be set in order to provide help for generated pages.

Table 7. Page Help Keys.

This table describes the property keys for generic Page Help properties.

Property	Description
Help.PageDescription.CreateEntity	Provides help for the generated create pages.
Help.PageDescription.List.EvidenceEntities	Provides help for all generated workspace list pages.
Help.PageDescription.ModifyEntity	Provides help for the generated modify pages.
Help.PageDescription.ViewEntity	Provides help for the generated view pages.
Help.PageDescription.List.EvidenceTypeVerifications	Provides help for the generated workspace verification pages.
Help.PageDescription.List.EvidenceTypeIssues	Provides help for the generated workspace issues pages.
Help.PageDescription.List.ChangeHistory	Provides help for the generated business object tab change history pages.

Field Label Keys

The following table shows the Field Label properties and their associated descriptions required in the general.properties file.

Each property can have an associated help property specified, by simply creating another property of the same name but with `.Help` appended.

Table 8. Field Property Keys.

This table describes the property keys for generic Field Label properties.

Property	Description
Field.Label.New	Label used for the `New` link in the actions menu on the generated evidence workspace.

Table 8. Field Property Keys (continued).

This table describes the property keys for generic Field Label properties.

Property	Description
Field.Label.Validate	Label used for hte `Validate` link in the actions menu on the generated evidence workspace.

Page Informational Keys

Page Informationals are warning messages shown on screen in response to user actions. These properties are a special case as they take the formatting options used in normal application message files and as such the number of parameters cannot be changed, however the message itself can be. No Help is required for these properties.

Page.Informational.NotModifiable: This warning message is shown on the screen when an end user attempts to modify an evidence entity record that was marked as not modifiable in the EUIM metadata

The suggested value for this is `%1s Evidence is not modifiable`

Table 9. Page.Informational.NotModifiable Parameters.

This table describes the parameters for the Page.Informational.NotModifiable property.

Parameter	Description
%1s	This parameter will be filled with the display name of the evidence entity in question

Static Properties

Action Control Label Keys

The following table shows the static Action Control properties and their associated descriptions required in the general.properties file.

Each property can have an associated help property specified, by simply creating another property of the same name but with `.Help` appended.

Table 10. Static Action Control Property Keys.

This table describes the property keys for generic static Action Control Label properties.

Property	Description
ActionControl.Label.Cancel	Button label used on multiple pages to cancel the action within that context.
ActionControl.Label.Close	Button label used on multiple pages to close the dialog.
ActionControl.Label.Save	Button label used in Create and Modify evidence entity pages to save new evidence entity.
ActionControl.Label.View	Link label used on to view specific evidence entity.

Table 10. Static Action Control Property Keys (continued).

This table describes the property keys for generic static Action Control Label properties.

Property	Description
ActionControl.Label.SaveAndNew	Button label used on Create pages to save and add a new entity of this type.
ActionControl.Label.Search	Button Label used for a Search button on various pages.
ActionControl.Label.Yes	Button label for a Yes button.
ActionControl.Label.No	Button label for a No button.
ActionControl.Label.New	Button label for a New button.
ActionControl.Label.Details	In Page Navigation link used on the view modal pages for an entity
ActionControl.Label.History	In Page Navigation link used on the view correction history modal pages for an entity
ActionControl.Label.Back	Label for a Back button, used on generated wizard screens
ActionControl.Label.Next	Label for a Next button, used on generated wizard screens
ActionControl.Label.Finish	Label for a Finish button, used on generated wizard screens

Field Label Keys

The following table shows the static Field Label properties and their associated descriptions required in the general.properties file.

Each property can have an associated help property specified, by simply creating another property of the same name but with `.Help` appended.

Table 11. Static Field Property Keys.

This table describes the property keys for generic static Field Label properties.

Property	Description
Field.Label.firstName	This label is used for the first name field when registering a new representative.
Field.Label.secondName	This label is used for the second name field when registering a new representative.
Field.Label.address	This label is used for the address field when registering a new representative.
Field.Label.areaCode	This label is used for the area code field when registering a new representative.
Field.Label.phoneNumber	This label is used for the phone number field when registering a new representative.

List Label Keys

The following table shows the static List Label properties and their associated descriptions required in the general.properties file.

There are no associated help properties.

Table 12. Static List Property Keys.

This table describes the property keys for generic static List Label properties.

Property	Description
List.Title.Type	This label is used for the type of an evidence object on an evidence create wizard screen.
List.Title.Description	This label is used for the description of an evidence object on an evidence create wizard screen.
List.Title.Period	This label is used for the period of an evidence object on an evidence create wizard screen.
List.Title.Participant	This label is used for the participant of an evidence object on an evidence create wizard screen.

Cluster Keys

The following table shows the static Cluster properties and their associated descriptions required in the general.properties file.

There are no associated help properties.

Table 13. Cluster Property Keys

Property	Description
Cluster.EvidenceHeader.Modify.Title	Title for the Cluster used to wrap the included infrastructure evidence header VIM on an entities generated modify screen.

Business Object Tab Keys

The following static properties are used when generating the business object tab for each evidence type.

There are no associated help properties.

Table 14. Business Object Tab Keys

Property	Description
leaf.title.Home	The title of the main navigation tab on generated Business Object Tabs for each entity type
leaf.title.ChangeHistory	The title of the change history navigation tab on generated Business Object Tabs for each entity type
leaf.title.Verifications	The title of the Verification navigation tab on generated Business Object Tabs for each entity type
leaf.title.Issues	The title of the Issues navigation tab on generated Business Object Tabs for each entity type
Submenu.Title.New	The actions menu New link for any child entities.. The display name of the child entity is automatically appended to the end of the property.

Table 14. Business Object Tab Keys (continued)

Property	Description
Submenu.Tooltip.New	The actions menu New tooltip for any child entities.. The display name of the child entity is automatically appended to the end of the property.
MenuItem.Title.Edit	
MenuItem.Tooltip.Edit	
MenuItem.Title.Delete	
MenuItem.Tooltip.Delete	
MenuItem.Title.ContinueEdititing	
MenuItem.Tooltip.ContinueEdititing	
MenuItem.Title.Discard	
MenuItem.Tooltip.Discard	
MenuItem.Title.CancelDeletion	
MenuItem.Tooltip.CancelDeletion	

Page Title Keys

The following static page titles are used in various pages created by the generator.

Note: ⃞ EvidenceFlow is deprecated. For more information, see Deprecated features.

Table 15. Static Page Title Keys

Property	Description
InPageNav.Label.Evidence	Title for the main list page of a generated evidence type workspace used within the evidenceFlow page.
InPageNav.Label.Verifications	Title for the verifications page of a generated evidence type workspace used within the evidenceFlow page.
InPageNav.Label.Issues	Title for the issues page of a generated evidence type workspace used within the evidenceFlow page.
Page.Title.NewEvidenceWizard	Page title for all generated create wizard pages.
Wizard.Text.SelectEmployment	Text for the wizard screen where a core Employment object is selected
Wizard.Title.SelectEmployment	Title for the wizard screen where a core Employment object is selected
Wizard.Text.SelectEvidence	Text for the wizard screen where a parent or pre associated Evidence object is selected
Wizard.Title.SelectEvidence	Title for the wizard screen where a parent or pre associated Evidence object is selected

Wizard Screen Description Keys

The following static properties are used as default, helpful text descriptions to users on the generated create wizard select screens.

The cluster for selecting a core employment record uses a simple text property, whereas the cluster for selecting a parent or pre association record uses a seperate starting and ending property, combined with a comma delimited list of the possible types being listed. For Example

For example, if the list contained records of type Paid Employment and Self Employment, the constructed description text would be as follows:

```
<Wizard.SelectEvidence.Description.Start> Paid Employment,
    Self Employment <Wizard.SelectEvidence.Description.End>
```

Table 16. Static Page Title Keys

Property	Description
Wizard.SelectEmployment.Description	Text description for the core Employment object list cluster on the generated create wizard pages.
Wizard.SelectEvidence.Description.Start	Start of the text description for the select evidence object cluster
Wizard.SelectEvidence.Description.End	Start of the text description for the select evidence object cluster

Sample File

Note: There should be no line breaks in individual properties.

```
##
##  Dynamic Values
##

##  Page Titles

Page.Title.EntityWorkspace=<displayName> Evidence

Page.Title.ModifyEntity=Edit <displayName> Evidence

Page.Title.NewEntity=New <displayName> Evidence

Page.Title.ViewEntity=View <displayName> Evidence


## Page Help Descriptions

Help.PageDescription.CreateEntity=This page allows you to create
  a <displayName> evidence record.

Help.PageDescription.List.EvidenceEntities=This page allows you
  to view a list of the <displayName> evidence recorded
    in the system.

Help.PageDescription.ModifyEntity=This page allows you to modify
  a <displayName> evidence record.

Help.PageDescription.ViewEntity=This page allows you to view a
  <displayName> evidence record.

Help.PageDescription.List.EvidenceTypeVerifications=This page
  allows you to view a list of the <displayName> verifications
    recorded in the system.

Help.PageDescription.List.EvidenceTypeIssues=This page allows
  you to view a list of the <displayName> issues recorded
  in the system.
```

```
Help.PageDescription.List.ChangeHistory=This page allows you to
  view the change history of a <displayName> record.


## Page Informationals

Page.Informational.NotModifiable=%1s Evidence is not modifiable


## Field Labels

Field.Label.New=New
Field.Label.New.Help=Press the New button to create a new
  <displayName> evidence record.

Field.Label.Validate=Validate
Field.Label.Validate.Help=Press the New button to create a new
<displayName> evidence record.

##
## Static Values
##

## Action Controls

ActionControl.Label.Cancel=Cancel
ActionControl.Label.Cancel.Help=Generic help message for cancel
  actions

ActionControl.Label.Close=Close
ActionControl.Label.Close.Help=Generic help message for close
  actions

ActionControl.Label.Save=Save
ActionControl.Label.Save.Help=Generic help message for save
  actions

ActionControl.Label.SaveAndNew=Save & New
ActionControl.Label.SaveAndNew.Help=The Save & New creates
  a new record from the information entered on the page and
  resets the page allowing an additional record to be created.

ActionControl.Label.View=View
ActionControl.Label.View.Help=Generic help message for View
  actions

ActionControl.Label.Search=Search
ActionControl.Label.Search.Help=Generic help message for search
  actions

ActionControl.Label.Yes=Yes
ActionControl.Label.Yes.Help=Yes

ActionControl.Label.No=No
ActionControl.Label.No.Help=No

ActionControl.Label.New=New
ActionControl.Label.New.Help=New

ActionControl.Label.Details=Details
ActionControl.Label.Details.Help=Shows details of the current
  record.

ActionControl.Label.History=History
ActionControl.Label.History.Help=Choose this to view the
  correction history of this record.
```

```
ActionControl.Label.Back=Back
ActionControl.Label.Next=Next
ActionControl.Label.Finish=Finish


## Field Labels

Field.Label.caseParticipant=Case Participant

Field.Label.participant=Participant

Field.Label.firstName=First Name

Field.Label.secondName=Surname

Field.Label.singleName=Name

Field.Label.address=Address

Field.Label.areaCode=Phone Area Code

Field.Label.phoneNumber=Phone Number

Field.Label.singleName=Name


## List Titles

List.Title.Type=Type

List.Title.Description=Description

List.Title.Period=Period

List.Title.Participant=Participant

## Page Titles

Page.Title.NewEvidenceWizard=New Evidence
Wizard.Text.SelectEmployment=Select Employment
Wizard.Title.SelectEmployment=Select Employment
Wizard.Text.SelectEvidence=Select Evidence
Wizard.Title.SelectEvidence=Select Evidence

InPageNav.Label.Verifications=Verifications
InPageNav.Label.Verifications.Help=Select this tab to view
  Verifications

InPageNav.Label.Issues=Issues
InPageNav.Label.Issues.Help=Select this tab to view Issues

InPageNav.Label.Evidence=Evidence
InPageNav.Label.Evidence.Help=Select this tab to view Evidence

## Generated Tab Properties

leaf.title.Home=Home

leaf.title.ChangeHistory=Change History

leaf.title.Verifications=Verifications

leaf.title.Issues=Issues

Submenu.Title.New=New
Submenu.Tooltip.New=New
```

```
## Miscellanious

Cluster.EvidenceHeader.Modify.Title=Change Details
Cluster.EvidenceHeader.Modify.Title.Help=Contains header details
   for the evidence record.

Wizard.SelectEmployment.Description=Please select one of the
   following Employments.
Wizard.SelectEvidence.Description.Start=Please select one of
Wizard.SelectEvidence.Description.End= from the following list.
```

employment.properties

The employment.properties file contains all generic label values for the employment pages generated. These consist of localized label values for all common buttons, page titles etc.

Note: The property keys cannot be changed, added or removed. To do so would cause errors in the running of the evidence generator.

Page Titles

The following table shows the Page Title properties and their associated descriptions required in the employment.properties file.

Table 17. Page Title Keys.

This table describes the property keys for generated Page Title properties.

Property Key	Description
Page.Title.Delete.Emploment	Title for the delete employment confirmation page.
Page.Title.Employment	Title for the employment list page.
Page.Title.Modify.Employment	Title for the modify employment page.
Page.Title.View.Employment	Title for the view employment page.

Field Labels

The following table shows the Field Label properties and their associated descriptions required in the employment.properties file.

Each property can have an associated help property specified, by simply creating another property of the same name but with `.Help` appended.

Table 18. Field Label Keys.

This table describes the property keys for generated Field Label properties.

Property Key	Description
Field.StaticText.CancelEmployment	Confirmation Text for removing an Employment
Field.Label.Primary	Label for the field indicating if this is a primary employment or not.
Field.Label.Occupation	Label for the field specifying the occupation associated with the employment.

Table 18. Field Label Keys (continued).

This table describes the property keys for generated Field Label properties.

Property Key	Description
Field.Label.Employer	Label for the field specifying the name of the employer.
Field.Label.From	Label for the field specifying the start date of the employment.
Field.Label.To	Label for the field specifying the end date of the employment.
Container.Label.Action	Label for the Action container field on generated pages.

Action Control Labels

The following table shows the Action Control Label properties and their associated descriptions required in the employment.properties file.

Each property can have an associated help property specified, by simply creating another property of the same name but with `.Help` appended.

Table 19. Action Control Label Keys.

This table describes the property keys for generated Action Control Label properties.

Property Key	Description
ActionControl.Label.Delete	Label for a `Delete` button.
ActionControl.Label.Edit	Label for an `Edit` button.
ActionControl.Label.Employment	Label for an `Employment` button.

Sample File

Note: There should be no line breaks in individual properties.

```
### Field Labels

    Field.StaticText.CancelEmployment=Are you sure\
      you want to delete this Employment?

    Field.Label.Primary=Primary

    Field.Label.Occupation=Occupation

    Field.Label.Employer=Employer

    Field.Label.From=From

    Field.Label.To=To

    Container.Label.Action=Action

    ### Page Titles

    Page.Title.Delete.Emploment=Delete Employment Details

    Page.Title.Employment=Employment

    Page.Title.Modify.Employment=Modify Employment Details
```

```
Page.Title.View.Employment=View Employment Details

### Action Controls

ActionControl.Label.Delete=Delete

ActionControl.Label.Edit=Edit

ActionControl.Label.Employment=Add Employment
```

General Error Messages

The general error message file for a component should be named
Ent<product.prefix>GeneralError.xml and be located in the components messages
folder.

Note: The <product.prefix> used above represents the same value as specified in
the property product.prefix in your components evidence.properties file.

See "evidence.properties" on page 12 for full details.

ERR_FV_CREATE_PROVIDER_DETAILS_SET_NO_NAME

This error message warns of a missing name field when other details were
provided for registering a new representative.

The suggested value for this is:
```
The %1s Name must be entered when any of the
    %1s details are entered.
```

where the argument is the case participant field being specified.

ERR_FV_CREATE_PROVIDER_NAME_SET_NO_ADDRESS

This error message warns when no address is specified while registering a new
representative.

The suggested value for this is:
```
The %1s Address must be entered when the
    %1s Name is entered.
```

where the argument is the case participant field being specified.

ERR_FV_FIELD_MUST_BE_ENTERED_WHEN_ANOTHER_FIELD_ENTER

This error message warns when one field is specified and another isn't when
registering a new representative.

The suggested value for this is:
```
The %1s must be entered when the %2s is entered.
```

where the argument values are the two fields in question.

ERR_FV_REMOVE_RECORD_ASSOCIATED

This error message warns when discarding an evidence record when it has an
associated record.

The suggested value for this is:

```
This %2s record cannot be discarded as there is an
        associated %1s record.
```

where the argument values are the types of the evidence records in question.

ERR_FV_NO_PARENT_RECORD

This error message warns of a missing parent record when creating a child record.

The suggested value for this is:

```
This %1s record cannot be discarded as the
        parent %2s does not exist.
```

where the argument values are the two evidence types in question.

ERR_FV_PARTICIPANT_EMPTY

This error message warns when no participant was chosen or a new one specified for a case participant field.

The suggested value for this is:

```
%1s Details must be provided.
```

where the argument value is the participant field left empty.

ERR_XFV_MORE_THAN_ONE_PART

This error message warns when more than one option is chosen for a case participant field i.e. a registered person is chosen and a new representative is specified as well.

The suggested value for this is:

```
Only one %1s can be entered. Please search for
        a registered %1s or enter details for an unregistered %1s.
```

where the arguments are the name of the field in question.

ERR_XFV_PHONE_NUMBER

This error message warns when an incomplete phone number is provided while registering a new representative.

The suggested value for this is:

```
Phone Number must be entered when Phone Area Code is entered.
```

ERR_XFV_PHONE_AREA_CODE

This error message warns when an incomplete phone number is provided while registering a new representative.

The suggested value for this is:

```
Phone Area Code must be entered when Phone Number is entered.
```

where the arguments are the name of the field in question.

ERR_XRV_CHILD_EXISTS_FOR_PARENT_TO_DISCARD

This error message warns when an attempt is made to discard a parent record which has a child record associated with it.

The suggested value for this is:

```
This %1c record cannot be discarded as there is a
    related %2c record. To discard the %3c record,
    you must first discard/remove the
    related %4c record.
```

where the arguments are the evidence types concerned.

ERR_FV_CASEPARTICIPANT_CHANGE

This error message warns when an attempt is made to change the case participant on an evidence record.

The suggested value for this is:

```
A participant cannot be changed for this evidence.
```

ERR_FV_EVIDENCE_SELECTION_REQUIRED

This error message warns when no parent or pre association record was chosen on the create new evidence wizard screens

The suggested value for this is:

```
An Evidence record must be selected.
```

ERR_FV_EMPLOYMENT_SELECTION_REQUIRED

This error message warns when no employment record was chosen on the create new evidence wizard screens.

The suggested value for this is:

```
An Employment record must be selected.
```

Server Meta Data

Document Structure

The server side meta data is provided as a well formed XML document, named <Entity Name>.xml. The following is the full reference for the structure of the xml file:

Entity Node (Required)

The Entity node is the root of the meta data document, and contains some basic information about the entity:

Table 20. Entity Node Attributes

Attribute	Mandatory	Possible Values	Description
logicalName	Yes	Any valid entity name	The logical name of the entity, as it appears on the database.

Table 20. Entity Node Attributes (continued)

Attribute	Mandatory	Possible Values	Description
relateEvidenceParticipant	No	Any valid case participant attribute from the entity	If set, this shows the participant to be set on the EvidenceDescriptor record. If left blank on a top level entity, the participant field on the descriptor will be set to the Primary Client of the associated case. If left blank on a child entity, the generator will iterate up the hierarchy (Parent, Grandparent etc) until a suitable participant is identified.

RelatedEntityAttributes Node (Required):

Table 21. RelatedEntityAttributes Node Attributes

Attribute	Mandatory	Possible Values	Description
relatedEntityAttributes	Yes	Yes / No	This is a Yes / No attribute. It determines whether or not the entity has related entity attributes. Related entity attributes are considered to be any piece of data required that cannot be read from the entity table directly. The result of the value 'Yes' is that an additional class is created with a method stub. The developer must handcraft the code required to read any related entity attributes.
exposeOperation	Yes	Yes / No	This is a Yes / No attribute. It determines whether or not the business process for retrieving the related entity attributes should be exposed to a façade thereby generating beans for it. This bean will also be used on the create screen.

Relationships Node (Required): This node is used to specify all relationship details about the entity. Entities can have 0..n relationships of type Parent, Child, Mandatory Parents, Pre Association or Related.

Parent Node (Optional)

The developer adds a Parent node for every possible parent type that the evidence entity has.

Table 22. Parent Node Attributes

Attribute	Mandatory	Possible Values	Description
name	Yes	Any valid evidence entity name	The logical name of the parent evidence entity.

Mandatory Parents Node (Optional)

In the case where an entity has multiple parents, which must all be specified, the <Parent> elements should be wrapped in an outer <MandatoryParents> element as follows:

```
<MandatoryParents>
  <Parent name="Parent1"/>
  <Parent name="Parent2"/>
</MandatoryParents>
```

Child Node (Optional)

The developer adds a Child node for every possible child type that the evidence entity has.

Table 23. Child Node Attributes

Attribute	Mandatory	Possible Values	Description
name	Yes	Any valid evidence entity name	The logical name of the child evidence entity.

PreAssociation Node (Optional)

The developer adds a PreAssociation node where the entity needs to be associated with another entity before creation so that related attributes from the associated entity may be displayed on the create screen

Table 24. PreAssociation Node Attributes

Attribute	Mandatory	Possible Values	Description
to	Yes	Any valid entity name	The evidence type that the entity is associated 'to'.

Related Node (Optional): The developer adds a Related node for every possible related type that the evidence entity has. The Related relationship is used to relate evidence entities to non-evidence entities.

Note: Note that currently the generator only supports relating to the core Employment entity, and as such that is the only option.

The Related Node can contain zero to many ParticipantType Nodes.

Table 25. Related Node Attributes

Attribute	Mandatory	Possible Values	Description
to	Yes	Any valid non evidence entity name	The entity that the entity is related to.

ParticipantType Node (Optional)

The ParticipantType node defines a case participant role type, to use when relating a non evidence record to the evidence one.

The system will find all participants registered on the case with the appropriate role type, and then list all records of the Related type for those participants, which can then be chosen to relate the evidence record to.

Table 26. ParticipantType Node Attributes

Attribute	Mandatory	Possible Values	Description
type	Yes	Any valid case participant role type	The case participant role types listed are used to relate the evidence entity type with the related, non-evidence entity. It facilitates navigating to the screen which lists the records the entity can be related to.

BusinessDates Node (Optional): The Evidence Interface now defines two methods, getStartDate and getEndDate that return the business dates of the entity, and are used in the period calculation.

The BusinessDates Node allows you to note which date attributes of the entity should be returned from these methods.

Table 27. BusinessDates Node Attributes

Attribute	Mandatory	Possible Values	Description
startDate	No	Any valid date attribute of the entity.	The date attribute to use as the business start date for the entity.
endDate	No	Any valid date attribute of the entity.	The date attribute to use as the business end date for the entity.

Override Node (Optional): This element should be used when a customer wishes to override / extend an out-of-the-box entity.

The relevant metadata should be copied to the custom evidence directory and, at a minimum, this element should be added.

This element should also be added where a new entity is being added to the product being overridden.

Note: For a full example of using the `Override` element please see the *Cúram Temporal Evidence Generator Cookbook*

Table 28. Override Node Attributes

Attribute	Mandatory	Possible Values	Description
newEntity	No	Yes / No	Shows if this is a completely new entity or not
customize	No	Yes / No	Set to Yes if you wish to override the provided Customise<Entity Name> class
hook	No	Yes / No	Set to Yes if you wish to override the provided <Entity Name>Hook class
relatedAttribute	No	Yes / No	Set to Yes if you wish to override the provided <Entity Name>RelatedEntityAttributes class
validation	No	Yes / No	Set to Yes if you wish to override the provided Validate<Entity Name> class

`CachedOperation` Node (Optional): The CachedOperation node allows the developer to specify a database read operation to be cached by the application. The developer is required to provide the name of the operation to be cached, the name of the database read operation and the fully qualified name of the return struct

Table 29. CachedOperation Node Attributes

Attribute	Mandatory	Possible Values	Description
operationName	Yes	Any sensible operation name	The name of the cached operation.
databaseRead	Yes	The name of the SQL entity read to be cached.	The name of the SQL entity read to be cached.
returnType	Yes	The qualified name of the struct that is returned by the entity read.	The qualified name of the struct that is returned by the entity read.

Client Meta Data

Document Structure

The client side meta data is provided as a well formed XML document, named <Entity Name>.euim along with associated properties files that can be specified in multiple locales. The following is the full reference for the structure of the xml file:

Entity node

Table 30. Entity Attributes

Attribute	Mandatory	Possible Values	Description
name	Yes	Any valid entity name	The logical name of the entity.
displayName	Yes	Any sensible String value	The name of the entity as it is to appear on client screens. For example, an entity may have a logical name of 'PaidEmployment' but on the client screens it is more desirable to display the name as 'Paid Employment'.
modify	No	Yes / No	This attribute shows whether or not the entity should be modifiable. This attribute is defaulted to Yes.

UserInterface Node

The UserInterface node is the beginning of the screen layout.

Table 31. UserInterface Node Attributes

Attribute	Mandatory	Possible Values	Description
saveAndNewButton	No	Yes / No	Defaults to No. If set to Yes, then a 'Save And New' Button will be added to the entity's create page.

Clusters Node

The Clusters node contains each individual cluster.

Cluster Node

The Cluster node contains information about each field that appears in the cluster. It can contain any number of Field or SkipField elements in any order.

Table 32. Cluster Node Attributes

Attribute	Mandatory	Possible Values	Description
description	No	Any valid entry from the associated properties file	This attribute maps directly to the UIM cluster description attribute.
numCols	No	Integer value	This attribute maps directly to the UIM cluster numCols attribute. If not specified the value defaults to 2.

Table 32. Cluster Node Attributes (continued)

Attribute	Mandatory	Possible Values	Description
label	No	Any valid entry from the associated properties file	This attribute maps directly to the UIM cluster TITLE attribute.
create	No	Yes / No	By default a cluster will be shown on the create page. To stop a cluster being shown on the create page set this attribute to No.
modify	No	Yes / No	By default a cluster will be shown on the modify page. To stop a cluster being shown on the modify page set this attribute to No.
view	No	Yes / No	By default a cluster will be shown on the view page. To stop a cluster being shown on the view page set this attribute to No.

SkipField Node

The SkipField node tells the generator to insert a blank UIM Field in this position. It allows for greater control over the formatting of the fields in the UIM Cluster.

Field Node

The Field node contains information about the attributes of a particular field on the screen.

Table 33. Field Node Attributes

Attribute	Mandatory	Possible Values	Description
columnName	Yes	A valid attribute name.	The entity attribute name, as it appears on the database.
label	No	Any valid entry from the associated properties file	This attribute maps directly to the UIM field label attribute.

Table 33. Field Node Attributes (continued)

Attribute	Mandatory	Possible Values	Description
modify	No	Yes / No / Many	By default a field is modifiable on the entity's modify screen. By setting this attribute to No, the field will be read-only on the modify screen. Case Participant fields are slightly different as they are typically not modifiable. By setting this attribute to Yes, it will be possible to leave it blank on the create and to set it once on the modify screen. After it has been entered once, it will be read-only on the modify screen. Alternatively, by setting this attribute to Many, it will be possible to overwrite it many times on the modify screen.
use_default	No	true / false	If specified, maps directly to the UIM field USE_DEFAULT attribute
use_blank	No	true / false	If specified, maps directly to the UIM field USE_BLANK attribute
notOnEntity	No	Yes / No	If set to Yes this attribute indicates that the field is not directly mapped to an entity field. Teh default is No
metatype	No	Any metatype recognized by the Evidence Generator (please see the appendix for a complete listing)	The metatype is a way of specifying additional information about an attribute, and how it should be formatted.

Table 33. Field Node Attributes (continued)

Attribute	Mandatory	Possible Values	Description
mandatory	No	Yes / No	Determines whether the mandatory indicator should be set on the field in the create and modify screens. The default is No

CaseParticipant Node

The CaseParticipant node contains additional information about the field relating to Case Participant information that is stored in the field.

Table 34. CaseParticipant Node Attributes

Attribute	Mandatory	Possible Values	Description
create	No	Yes / No	Determines whether a Case Participant is to be created.
name	No	Any sensible String value.	The name refers to how the associated attribute has been named. For example, the attribute may be named 'empCaseParticipantRoleID', in this instance the name attribute must have a value of 'emp'.
readOnly	No	Yes / No	Determines whether the Case Participant is 'read only'.

Table 34. CaseParticipant Node Attributes (continued)

Attribute	Mandatory	Possible Values	Description
nsStruct	No	Yes / No	When searching for a previously registered participant on the system, the default is to search for the `Person` type. In order to search for other types, this should be set to `Yes` and the developer should model their own struct with the same structure as `CaseParticipantDetails`. and aggregate this instead. A further attribute called `participantType` should be added, that links to a codetable of participant types that should be searchable. This will then give the end user a drop down list of participant types beside the search button, and the correct search dialog will appear based on the type selected from this list.
singleNameField	No	Yes / No	This attribute is used when a newly registered participant for this field should only have one name rather than a first and second name e.g. if registering a School.

CreateCaseParticipant Node

The `CreateCaseParticipant` node contains information about creating a Case Participant. Including this node indicates that the chosen participant should be registered as the specified case participant type if they are not already registered as such.

Table 35. CreateCaseParticipant Node Attributes

Attribute	Mandatory	Possible Values	Description
participantType	Yes	Any valid Participant type from Appendix D	This field gives more meta information to the generator about the type of Participant being dealt with
roleType	Yes	The Java Identifier of an entry from the CaseParticipantRoleType codetable	Specifies the role that the participant should be registered on the case with.

SearchType Node

When using a case participant field, the system can provide a pre populated, drop down list of existing case participants that can be chosen from. The `SearchType` node, which can be specified multiple times within a `CaseParticipant` node, indicates which case participant role types to include in this list.

Table 36. SearchType Attributes

Attribute	Mandatory	Possible Values	Description
type	Yes	Any valid Java Identifier from the CaseParticipantRoleType codetable.	This shows which Case Participant Role Type to list for selection.

Metatypes

Currently the following metatypes are supported by the Evidence Generator:

PARENT_CASE_PARTICIPANT_ROLE_ID

On Child or Grandchild evidence, applying this metatype to a field causes that field to display as the name of the parents associated case participant role (e.g. James Smith) as well as being a link to the Participant Home page.

EMPLOYER_CASE_PARTICIPANT_ROLE_ID

Applying this metatype to a field implies that the field is storing an Employer's participant role ID in the field. The result is that the Employer's name is displayed as a link to the Employer Home Page.

CASE_PARTICIPANT_SEARCH

Applying this metatype to a field implies that the field is storing the case participant ID of the case participant that the evidence record is being associated with. The result is that the participant's name is displayed as a link to the Participant Home Page.

CODETABLE_CODE

Applying this metatype to a field implies that the field is storing a codetable value which is to be displayed as part of the description String generated by the `StandardEvidenceInterface::getDetailsForListDisplay` function. The result is that the code stored in the field is replaced by the description String from the codetable.

REPRESENTATIVE_LINK

Applying this metatype to a field implies that the field is storing an ID which can be used to link to the Representative Home Page. The result is that the Representative's name appears as a link to the Representative Home Page.

COMMENTS

Applying this metatype to a field implies that the field is storing free-text. The result is that the field will be the full width of the screen and three rows in height.

RELATED_ENTITY_ATTRIBUTE

Applying this metatype to a field tells the generator that the field actually comes from the modelled related attributes struct rather than the entity itself. Fields of this type are read-only.

Participant Types

Currently the following Participant Types are supported by the CreateCaseParticipant node in the evidence generator:

Note: The closest match to the participant type to be created should be chosen

- Person
- Employer
- ServiceProvider
- Union
- Unknown

Notices

This information was developed for products and services offered in the United States.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM® product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing IBM Corporation North Castle Drive, MD-NC119 Armonk, NY 10504-17

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing Legal and Intellectual Property Law IBM Japan Ltd. 19-21, Nihonbasl

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

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

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Director of Licensing IBM Corporation North Castle Drive, MD-NC119 Armonk, NY 10504-1785

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

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

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

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

COPYRIGHT LICENSE:

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

Privacy Policy considerations

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings

can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

Depending upon the configurations deployed, this Software Offering may use session cookies or other similar technologies that collect each user's name, user name, password, and/or other personally identifiable information for purposes of session management, authentication, enhanced user usability, single sign-on configuration and/or other usage tracking and/or functional purposes. These cookies or other similar technologies cannot be disabled.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM's Privacy Policy at http://www.ibm.com/privacy and IBM's Online Privacy Statement at http://www.ibm.com/privacy/details the section entitled "Cookies, Web Beacons and Other Technologies" and the "IBM Software Products and Software-as-a-Service Privacy Statement" at http://www.ibm.com/software/info/product-privacy.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at " Copyright and trademark information " at http://www.ibm.com/legal/copytrade.shtml.

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.

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

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

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

IBM ®

Printed in USA