Merative Social Program Management 8.1

Configuration Transport Manager guide

Note

Before using this information and the product it supports, read the information in

Edition

This edition applies to Merative™ Social Program Management 8.0.0, 8.0.1, 8.0.2, 8.0.3, and 8.1.

Contents

1 Cúram Configuration Transport Manager overview

The Configuration Transport Manager enables administrators to migrate application data and configurations between systems as business objects. Use this information to understand what data can be transferred and how to set up a data transfer. A business object type is a logical grouping of administrative data that defines and governs a particular set of functions.

A typical system landscape consists of a development system, a test system, and a production system. The Cúram Business Application Suite includes a wide breadth of administration and configuration data. Customers typically need to configure and test this on development and test systems ahead of deploying to production. The Configuration Transport Manager provides the functionality to transport data across the system landscapes and ensure the data is fully tested, reliably transported, and applied to each system, while also retaining the integrity of data objects.

1.1 Setting up the Configuration Transport Manager

To set up systems to use Configuration Transport Manager, you must complete the prerequisites. Then you can set up Configuration Transport Manager on new application installations or you can enable the use of Configuration Transport Manager on existing application installations. You can also complete some other administrative activities on the systems where Configuration Transport Manager is in use.

Prerequisites for setting up the Configuration Transport Manager

The prerequisite activities that must be completed before setting up and using the Configuration Transport Manager.

Defining the system landscape
To set up and use the Configuration Transport Manager, the initial structure of the system landscape must first be defined. Use this information to learn about the minimum system landscape configuration that is needed and other system landscape requirements, such as ensuring system homogeneity and configuring a unique landscape name.

Recommended minimum configuration

At a minimum, a system landscape must consist of three systems, one for each of the following roles: Development, Test, and Production. The following activities must be carried out on each of the systems in respect of the minimum configuration.

- **Development system**
  The development system is the system on which business objects are initially created and subsequently changed. When the Administrator is satisfied that the business objects are configured correctly, they are transferred and applied to the test system by using the Configuration Transport Manager.
- **Test system**
  The test system is the system on which business objects are tested before being transported and applied to the production system. The test system should contain a representative set of the runtime data from the production system. This is so that issues related to the interactions

between the business objects and the runtime data set are discovered and remedied before the business objects are transported and applied on the production system. To avoid consistency issues, do not create or change business objects on the test system. For information on policies for creating and changing business objects, see Data policy management. When the business objects are verified and behave as expected, they can be transferred and applied to the production system.

- **Production system**
  The production system is the live system that is used by end users. Business objects should only be transferred and applied on the production system after passing extensive testing on the test system. Great care must be taken before deciding to create or change a business object on the production system. For more information on policies for authoring and changing business objects, see *Defining a data management policy*.

Other system landscape considerations

Depending on need, other system landscape configurations are also possible. For example, a configuration with four systems, a development system, a test system, a UAT system, and a production system might be required. Alternatively, a development system, a test system, a training system, and a production system might be required.

At a minimum, every system landscape configuration should have at least one development system and at least one test system, both separate from the production system. It is also recommended that the number of systems in a system landscape is not allowed to grow too large. Otherwise, problems can arise in managing the transport paths between the systems, and associated problems can arise in respect of system consistency.

Ensuring system homogeneity

The systems in a system landscape must be as homogeneous as possible. This is to prevent issues arising due to system configuration differences.

For example, transporting a business object from a system that supports one set of locales to a system that supports a different set of locales can cause problems when applying or using the business object.

To this end, all systems in a system landscape should be the same with respect to at least the following:

- **Version**
  All systems must use the same version of the application.
- **Customer customizations**
  All systems must contain the same customer-developed customizations.
- **Third-party software**
  All systems must use third party software from the same vendor, and must use the same version of the third party software. For example, the application server, the database, etc. must all be exactly the same on each system.
- **Time zone**
  All systems must all be in the same time zone.
- **Locale**
  All systems must support the same locale or locales.
- **Date format**
  All systems must use the same date and date-time format.

- **Database character set**
  The databases on all systems must use the same character set.

Defining a system landscape name

A system landscape must have a unique name. Each system in the system landscape is configured with this name. So, some thought needs to be put into determining an appropriate name for the system landscape. For example, the system landscape name might be the organization name.

- The landscape name should consist only of alphanumeric characters, with no spaces.
- The maximum landscape name length is 500 characters, but it is recommended that a smaller name is used.
- Do not name the landscape *nolandscape*. For more information, see *nolandscape* in Landscape name.

Defining multiple system landscapes

Optionally, it is possible to set-up multiple system landscapes, each containing different, separate systems. For example, this may be desirable if trialling a new version of the application.

However, note that each system can only be in one system landscape, and that once assigned to a system landscape, a system cannot be moved to another system landscape. Furthermore, business objects should not be transported between systems in different system landscapes.

Defining a data management policy

Define a management policy for managing administrative data within the system landscape. Key elements include defining a business object creation and modification policy, deciding when to apply change sets in production, defining a test system data management policy, assigning key block ranges, and managing cryptography data.

Business object creation and modification policy

An important element of the data management policy is to decide on the procedure that is used to create and modify business objects. There are two basic options.

- **Option 1: Use standard development lifecycle practices for all business object types**
  The preferred option is to manage all business objects by using standard development lifecycle practices. This means creating and modifying business objects only on the development systems in a system landscape. With this approach, no business objects are created or modified on test or production systems. Instead, the new or modified business objects are transported from the development system to the test system and following testing, transported to the production system. This help to ensure that there are no issues with a particular business object, removes the potential for transported business objects to overwrite local changes on production systems, and reduces clashes that can prevent the application of a change set.
- **Option 2: Manage some business object types locally**
  In some cases, option 1 might not be sufficiently flexible. For example, it might be necessary to make changes to some business objects directly on the production system, where the danger of issues arising is low, and where the overhead of creating business objects on a different system, and then transporting them, is too high.

  In this case, a clear distinction must be made between business object types that are managed locally, and those that are managed by using standard development lifecycle practices, with Configuration Transport Manager. Use Configuration Transport Manager only with business

objects whose business object type is managed by using standard development lifecycle practices. Do not use Configuration Transport Manager to transport business objects whose business object type is managed locally on the production system.

Applying change sets to a production system

Decide when change sets can be applied to the production system. It is recommended that change sets are applied only to the production system during scheduled maintenance time when the system is offline. This allows for controlled verification and validation of the changes made to the production system by the change set without affecting end-users. By doing this, the potential for negative impact from unexpected issues arising due to the changes is reduced.

Defining a test system data policy

The test system is used to test business objects before they are transported and applied on the production system. Therefore, it is important that the business object data on the test system is as close as possible to that on the production system. If there are substantial differences between the business objects on the test system and those on the production system, the validity of the tests performed on the test system will be compromised.

Two procedures can lead to divergence between the test system business objects and the production system business objects.

1. If change sets that contain business objects are transported and applied on the test system, but never transported and applied to the production system over a period of time, the business objects on the test system will diverge from those on the production system.
2. If new business objects are added or changes are made to business objects directly on the production system, then the test system and production system configurations will differ.

> **Note:** Adding or changing business objects directly on the production system is strongly discouraged.

To avoid substantial differences between the test system and production system, it is recommended that customers adopt a policy to ensure that the business object data on the two systems is kept in sync. It is suggested that the following elements form part of this policy:

1. When the set of business objects in a change set passes testing on the test system, ensure that they are transported and applied on the production system as soon as possible. That is, as soon as a scheduled maintenance window is available.
2. If the business objects fail testing on the test system, ensure that the issues with the business objects are addressed on the development system as soon as possible, and then retested on the test system. This help to ensure that the test system does not contain invalid configurations that cannot be migrated to the production system.

Managing key block range allocations

A unique key block range must be assigned to each system in the system landscape to avoid clashes in primary keys when a business object is transported and applied. For more information, see .

It is very important that no systems in the landscape have duplicate key block range assignments. Therefore, the key block range assignment to systems should be managed centrally, by a single authority within an organization.

Managing cryptography data

Cryptographic configurations can be upgraded. Therefore, you need to be aware of the configuration data that may differ between source and target systems.

By default, this data is restricted to the following areas.

- **Passwords in the Users table**
  Subject to digest cryptographic setting differences across systems.
- **Passwords in the ExternalUser table**
  Subject to digest cryptographic setting differences across systems.
- **Passwords in the TargetSystemService table**
  Subject to digest cryptographic setting differences across systems.
- **Properties in the Properties table (using `Application.prx`) that contain password values**
  Subject to cipher cryptographic setting differences across systems.

If there are source and target differences in the cryptographic configurations, you must re-digest or re-cipher, or re-digest and re-cipher, the password data. For information about the Apache Ant targets for doing this, see the *Cúram Server Developer's Guide*.

Enabling Configuration Transport Manager on a new system

Learn about the tasks that you must complete to set up a new system with Configuration Transport Manager enabled.

Configuring landscape names

You must configure a landscape name for each system in the system landscape. Users can transport and apply change sets only to other systems that are configured with the same landscape name. If no landscape name is configured for a system, by default, a landscape name of nolandscape is applied.

Configure the landscape name by using the Configuration Transport Manager setup tool. For more information, see .

Default landscape name restriction

There is a restriction on the use of Configuration Transport Manager on systems with the default landscape name nolandscape. You cannot create change sets that are created on systems with the nolandscape name to any other system, including other systems configured with the nolandscape name. To make full use of the capabilities in Configuration Transport Manager, you must explicitly configure each system with a different landscape name than the default, nolandscape.

Configuring key block ranges

The key block range ensures that when a business object is transported to another system, the primary keys, and unique IDs that used by its entities, do not conflict with the entity IDs on the target system. You must assign a unique key block range to each system in the system landscape.

Before you begin

See the key block range information in for information about how key block ranges are used in the application. For more information about how unique IDs are used in the application, see .

> **Note:** If a single system uses the default key block range allocation *group 3, range 2*, you cannot configure any other systems in the system landscape with the key block range group 3, range 2. If multiple key block ranges are allocated to a system, the additional key block range allocations must also be unique to the system and you cannot use them on other systems within the system landscape.

Specifying the group name and group number

You must configure each system in the system landscape with a unique group range pair. For example, you might use the following configuration for a three system landscape:

- **Production system**
  Group 3, Range 2
- **Test system**
  Group 3, Range 3
- **Development system**
  Group 3, Range 4

Alternatively, you might assign the following ranges:

- **Production system**
  Group 3, Range 2
- **Test system**
  Group 4, Range 2
- **Development system**
  Group 4, Range 3

Setting up the key block range allocation for each system

Set up the key block range allocation for each system in the landscape by configuring values in the Extensible Markup Language (XML) configuration file, and by using the Configuration Transport Manager setup tool. For more information, see Configuring the landscape name and key block range allocation.

Configuring the landscape name and key block range allocation

Use the Configuration Transport Manager set up tool to configure both the landscape name and the key block range allocation for each system in the system landscape. Specify the appropriate values in the Extensible Markup Language (XML) configuration file. Then, start the setup tool to write the configured values to the database.

The default configuration file location is `<CuramSDEJ>/scripts/config/rangeawareserver_config.xml`. You can edit the values in this file directly, or you can copy the file to an alternative location and edit it there. If you use an alternative location, you must specify the location when you execute the setup tool.

> **Remember:** Configure the values and execute the tool before the system is started.

Example configuration: Landscape name

Use the `<Landscape>` stanza in the configuration file to specify the landscape name. The default value for the landscape name is *nolandscape*. However, you can change this to the name of the landscape.

```
<Landscape name="landscapename">
        <value>mylandscape</value>
        </Landscape>
```

Example configuration: Key block range allocation

Use the `<Rangevalue>` stanza to configure the key block range allocated to the system. The key block range consists of two elements: a group number and a range number. You can specify both in the `<Rangevalue>` stanza.

```
<Rangevalue>
        <attribute name="groupNumber">
        <value>3</value>
        </attribute>
        <attribute name="rangenumber>
        <value>5</value>
        </attribute>
        </Rangevalue>
```

The configuration file can contain more than one `<Rangevalue>` stanza, with the same structure used for each of the stanzas. This functionality is used to specify multiple key block range allocations if required.

Note that both the group number and the range number are validated by the tool to ensure that the values are acceptable.

Starting the Configuration Transport Manager setup tool

The setup tool is an Apache Ant® target. You can run the target as part of the standard Ant **database** target, and you can run it independently. First, set the appropriate values for the landscape name and the key block range allocation in the configuration file, then perform the database build. The configured values are loaded into the system as part of the database build. That is, they are loaded by executing **build database**.

Alternatively, you can load the configured values into the database separately, by invoking the **insertrangeawareconfig** target, that is, by running **build insertrangeawareconfig**.

If an alternative location to the default location is used for the configuration file, you can specify this by setting the system property `rangeAwareConfig` when invoking Ant. For example, you can run the **database** target as follows:

```
build database -DrangeAwareConfig=<config-file-path>/rangeawareserver_config.xml
```

*Figure 1: Specifying an alternative configuration file location when initiating the **database** target*

You can run the **insertrangeawareconfig** target by using:

```
build insertrangeawareconfig
 -DrangeAwareConfig=<config-file-path>/rangeawareserver_config.xml
```

*Figure 2: Specifying an alternative configuration file location when initiating the **insertrangeawareconfig** target*

Configuring the system identifier

You must assign a unique human-readable name known as the Configuration Transport Manager system identifier to each system in the landscape. The Configuration Transport Manager user interface uses the system identifier to indicate the source system on which a change set originated.

There are some restrictions on the value that you select for the system identifier:

- The system identifier must consist only of alphanumeric characters, with no spaces.
- The maximum length of the system identifier is 500 characters, but it is recommended that you use a smaller identifier.
- The system identifier should not be named *noname*.

Default system identifier

If you do not configure a Configuration Transport Manager system identifier, a default value of *noname* is used. Systems that use the default *noname* value are restricted in that change sets created on these systems cannot be released. This means that to make full use of the capabilities offered by Configuration Transport Manager , the system must be explicitly configured with a Configuration Transport Manager system identifier that differs from the default *noname* value.

Setting the *curam.ctm.systemIdentifier* property

You configure the system identifier by using the property *curam.ctm.systemIdentifier*. This property is set using the system administrator *Property Administrator* functionality in the application. The property is contained in the *Admin - System Settings* category. After you set the value, you must publish it to make it available to the system.

> **Note:** It is highly recommended that once set, this property value is not changed again. Changing the property may cause confusion with regard to the origin of transported change sets when they are viewed on a target system.

Enabling Configuration Transport Manager on an existing application

A customer might want to start using the application in production without initially making use of Configuration Transport Manager. This strategy avoids the need to initially consider Configuration Transport Manager and specific concerns, such as system landscape configuration, data management policies, and key block range allocation.

Consideration of these issues can be deferred until the decision is made to start using Configuration Transport Manager.

An example of when this strategy might be adopted is when an existing customer initially upgrades to a distribution of the application that supports Configuration Transport Manager. The customer might decide to first perform a technical upgrade, avoiding use of new features introduced in the new version of the application. Then, at a later date, the customer may decide to commence using new features such as Configuration Transport Manager.

Configuration Transport Manager restrictions

If a system is installed and started without completing the setup steps for the Configuration Transport Manager, it has the default landscape name *nolandscape*, the default key block range allocation *group 3, range 2*, and the default system identifier *noname*. With these default values,

the Configuration Transport Manager can only be used in a restricted form. To allow for full use of all features, a setup procedure must be implemented.

The restriction means that change sets created on the system cannot be transported and applied on any other system. Therefore, a production system that is set up without completing the Configuration Transport Manager setup steps allows only for usage of Configuration Transport Manager subject to these restrictions.

It is recommended that a setup procedure is first trialled on a non-live duplicate of the production environment before it is carried out on the production environment. All of the activities that involve making changes to the production system should be carried out during scheduled maintenance time, when the production system is offline.

Prerequisites for enabling Configuration Transport Manager on an existing application

Before starting, a number of prerequisite activities must be carried out. These are in addition to the prerequisites for setting up Configuration Transport Manager.

Table 1: Prerequisite checklist for enabling Configuration Transport Manager on an existing application

Prerequisite	Description
Decide on a system landscape name	The landscape name must be defined. For more information, see Defining a system landscape name on page 11.
	The production system is initially configured with the default landscape name (*nolandscape*). You must change this. For more information, see Setting up the Production System.
Decide on key block range allocations	The production system has the default key block range allocation of *group 3, range 2*. However, every other system in the same system landscape as the production system must also have a unique key block range allocation.
	For more information, see Managing key block range allocations on page 12.
Decide on Configuration Transport Manager system identifier	Each system must have a unique Configuration Transport Manager system identifier. The production system is initially be configured with the default Configuration Transport Manager system identifier (*noname*), but you must change this to enable Configuration Transport Manager.

Setting up a production system

To enable usage of Configuration Transport Manager on an existing production system, you must make two configuration changes. You must change the landscape name and set the Configuration Transport Manager system identifier for the production system.

Changing the landscape name

The landscape name on the production system must be changed from *nolandscape* to the new name that has been selected. Changing the name makes the production system a member of the landscape and enables full use of all Configuration Transport Manager functionality without restrictions.

> **Note:** Changing the landscape name on the production system should only take place during scheduled maintenance time, when the production system is offline.

To change the landscape name, change the value of the property *curam.ctm.landscape.name* from *nolandscape* to the selected landscape name. You can change this by using the system administrator *Property Administrator* functionality in the application. The property is contained in the *Admin - System Settings* category. After you change the value, publish the changes so that they are made available to the system.

> **Important:** Do not change the landscape name other than when performing this procedure to enable Configuration Transport Manager usage on an existing production system. On all other systems, the landscape name should never be changed from the value to which it is initially set.

Changing the production system identifier

You must change the system identifier for the production system from the default value of *noname* to the new value that is selected. For more information, see .

Setting up other systems in the system landscape

You must also set up any other systems in the system landscape that is, the development system, test system, and any other systems required. The systems must be fresh installations of the application with Configuration Transport Manager fully enabled at install time.

Installing the application

The other systems in the system landscape will require fresh application installations. For example, if the system landscape consists of a development system, a test system, and a production system, then the application needs to be installed on the development system and the test system.

Note that it is important that the systems in the system landscape are as homogeneous as possible. So, the other systems in the system landscape should match the configuration and setup of the existing production system as closely as possible. The installation must also include any customer customizations that are developed and applied to the application. For more information, see .

Loading initial data on the other systems

The other systems in the system landscape must initially contain the same business object data as the production system. That is, the administrative configuration data on the other systems in the system landscape must start out as the same as that on the production system. This means that changes can be made to preexisting business objects on the other systems in the system landscape and any testing carried out on the development system or test system is valid.

The production system data for the entities that form the business objects must be copied over to the other systems in the system landscape before the other systems are first started. These entities are listed in . It is recommended that you make the copy during scheduled maintenance time, when the production system is offline. This helps to ensure that no new business object data is created on the production system before the other systems are started.

Additional setup steps

After performing the above steps, the steps to setup Configuration Transport Manager for all new application installations must be implemented. These are to set up the systems with the selected

landscape name, set up the key block range allocations, and so on. For more information, see .

Other administrative considerations

This section provides information about other administrative activities that may be carried out on the systems where Configuration Transport Manager is in use.

Adding systems to an existing system landscape

It is possible to add more systems to an existing system landscape. However, these systems must use fresh installations of the application. That is, it is not possible to move an existing application installation into a different system landscape.

The steps involved in adding a new system to an existing system landscape are the same as those described in *Setting up other systems in the system landscape*.

> **Note:** Remember that the key block range that you allocate to the new system must not have been previously assigned to an existing system in the system landscape.

1.2 The change set lifecycle

A change set is the mechanism that administrators can use to package configuration data in the form of business objects and transport it from one system to another. Administrators can also use change sets to revert to a defined set of business objects at a later point in time. A change set goes through multiple stages in its lifecycle.

An administrator first creates a change set and then adds the configuration data that needs to be transported to the change set. When the contents of the change set are finalized, the administrator releases the change set. When the change set is released, the administrator transports the change set and configuration data to another system. The administrator then applies the change set in the system, which means that the configuration data is written to the database and becomes live on that system.

The following sample scenario further illustrates how the Configuration Transport Manager works. An administrator creates or changes some configuration data on a development system. They use the Configuration Transport Manager to transport the configuration data to a test system, which has a configuration that closely mirrors the production system. When the configuration data is successfully tested on the test system, the administrator uses the Configuration Transport Manager to transport the configuration data to the production system and make it live.

Defining a change set

Defining a change set consists of four stages. The stages are creating a change set, adding business objects and related business objects to the change set, and releasing the change set.

Creating a change set

The first step in the change set lifecycle is to create the change set. Created change sets have a status of 'Open'. When the change set is in 'Open' state, the business objects that are required for transport can be added to the change set.

Adding business objects to a change set

A business object is a set of administrative data that defines and governs an instance of a particular set of functionality. It is through the capture, transport, and application of sets of business objects that Configuration Transport Manager provides its capabilities. Each business object consists of the collection of data, that is, entity instances, that is required to configure the system to use or act on the instance of the functionality that it represents.

All of the business objects that contain configuration data suitable for transporting have been analyzed and integrated with Configuration Transport Manager. An administrator can select an instance of a business object and add it to a change set. The administrator does not require any further knowledge of the underlying data structure for the business object. For more information about the business objects, see .

When the change set is in an Open state, a user or a group of users can identify the business objects that need to be transported to a target system and add them to the change set. A search feature is provided so that users can search for business objects and add them. Business objects can also be removed from the change set when the change set is in an Open state.

Adding related business objects to a change set

Based on a business object already in the change set, related business objects are presented to the user and the user can determine if these should be added to the change set also.

The Add Related Business Objects feature helps to ensure data integrity on the target systems by assisting customers in transporting the correct changes. The business object and its related business objects are required to maintain the business object configuration data and relationships in the target system exactly as they exist in the source system. This helps to avoid application issues when the business object is used in the target system. If related business objects already exist on the target system, it may be unnecessary to transport them. Likewise if the change to be transported is a smaller one known to be localized to a particular business object, then it is fine to transport the single business object without its related business object. As there are many reasons for not transporting all related business objects, this feature provides guidance to users on what they may need to transport.

The Add Related Business Objects feature evaluates related business objects through database relationships, such as foreign key relationships or link tables. If any other type of relationship exists between business objects, it is not recognized by the Add Related Business Objects feature. For example, an API exists for Rate Table functionality and calls to this API from the codebase may introduce a relationship between a rate table and the calling business object. However, this relationship is not presented to the user as a dependent business object. The Add Related Business

Objects feature should only be considered a tool to provide guidance about what needs to be transported.

Releasing a change set

When an administrator releases a change set, the change set definition is complete and any business object data is frozen in the change set at this point. This is done when all the business objects that need to be transported as part of the change set have been added to the change set and no further changes are required. The change set contents, including the business object instances, cannot be modified once the change set is released. Any changes made to the business object after the change set is released will not be in the change set. When released, the administrator can transport the change set to a target system. They can also use the released change set to revert to the configuration data that is associated with the business objects in the change set at a later time.

Transporting business objects

The Configuration Transport Manager provides transportable business objects to package the configuration data that supports a business object in a change set. Administrators can automatically or manually transport the business objects in a released change set to a target system. To make the configuration data in the business objects live in the database, Administrators must apply the released change set. The Configuration Transport Manager also allows for bi-directional transport of configuration data between systems based on business needs.

Transporting a change set

Applying a released change set makes the configuration data in the business objects in the change set live in the database. To do this, CTM uses web services.

Administrators can manually transport released change sets to a target system. Use manual transport when it is not possible to create a connection between two systems over a network. For example, there might be network issues or security policies might mandate that the production systems must be hosted on networks that are isolated from development systems.

To manually transport a change set, three steps are required: export, copy and import. The change set is first exported from the source system. This downloads the change set, including the business objects, to the source system local drive from CTM. Then, the exported file is manually copied to the target system, for example, by e-mail, USB key, shared directory, etc. The file is then imported into the target system. The CTM import function reads the change set and all of its contents from the external file into the target system and creates it on that system in a released state. The change set is then ready and available to be applied on the target system.

For information about how to configure target systems for web services, see .

Applying a change set

Applying the change set can result in the addition of business objects in the target system if they did not already exist, or the modification of business objects in the target system if they already existed. When the applied changes are successfully tested in the target system, the change set can be transported to another system. For example, a change set is transported from a development system to a test system. The change set is applied in the test system and the underlying configuration data changes are successfully made live. If the configuration

data changes are successfully tested on the test system, the change set can be transported to a production system.

Administrators can also create and apply a change set in the same system so that they can revert to earlier working configuration data if application issues are discovered. For more information, see .

Bi-directional transport

CTM allows bi-directional transport of configuration data between systems in the system landscape. For example, if a customer ordinarily uses CTM for transporting their configuration changes from development through testing into production, they might need to apply a change outside this process directly on the production database. With the bi-directional support in CTM, the customer can re-sync the configuration data across systems in this scenario and use CTM to move the change from production, to testing, to development.

Cloning and deleting change sets

Administrators can clone a change set and then modify its content to meet changing requirements. A change set can also be deleted at any stage.

Cloning a change set

Administrators can clone a change set. However, the new version of the change set must have a new name for traceability purposes. A cloned change set has a status of Open and contains the list of business objects that existed in the change set that was cloned. Administrators can adder remove business objects from the change set as required. This helps administrators to automatically create a change set with business objects similar to an existing change set rather than creating a change set manually and then adding all of the required business objects.

When a change set is cloned, only the identifiers of the business objects are cloned. The data that is contained in the business objects is not cloned. When the new change set is released, it contains changes that were made to the business objects up to the point when the new change set was released.

Therefore, if any changes are made to the business objects between the point in time when the original change set was released, and the point in the time when the new cloned change set is released, the content of the business objects in the change sets will be different.

Deleting a change set

A change set can be deleted at any point in time. Operations cannot be performed on the change set once the change set is deleted. Deleting the change set changes the status of the change set to Canceled.

1.3 Troubleshooting Configuration Transport Manager errors

Using the Configuration Transport Manager often involves modifying configuration data in production systems. Therefore you must ensure that you protect the production system. A number

of features are provided to assist customers in protecting the production system or recovering the system if things go wrong.

Recommended usage

The Configuration Transport Manager is used to transport configuration data changes between systems in a structured manner and leads to the modification of configuration data in a production system. It supports the testing of configuration changes on development and test environments before they are deployed on a production system.

> **Note:** It does not remove responsibility from the administrator from understanding what they are transporting and when it should be transported.

For example, if there is a rate change for a particular benefit, but the particular rate table change is accidentally not brought from the test system to the production system, this could result in incorrect eligibility output. Another example involves two successive changes that are made to a business object and included in separate change sets, but the change sets are applied in reverse order. This might cause changes in the initial change set to be overwritten and not applied to the production system.

Given the examples of how Configuration Transport Manager can be used incorrectly, it is important that customers incorporate the use of Configuration Transport Manager into their own release process to give due attention to the contents of a change set and to ensure the correct and timely activation of a change set on the production system.

Even with the tightest release process, things can still occasionally go wrong. For more information about the recommended procedures for managing the creation and modification of business objects, see Defining a data management policy on page 11.

Recommended usage of Configuration Transport Manager

Configuration Transport Manager is used to transport configuration data changes between systems in a structured manner and often leads to the modification of configuration data in a production system. Therefore, it is important for Administrators to understand what they are transporting and when it should be transported, and that customers incorporate the use of Configuration Transport Manager into their release process.

For example, if there is a rate change for a particular benefit, but the particular rate table change is accidentally not brought from the test system to the production system this could result in incorrect eligibility output. Another example is where there are two successive changes made to a business object and included in separate change sets but the change sets are applied in reverse order. This could cause changes in the initial change set to be overwritten and not applied to the production system.

Given the examples here of how Configuration Transport Manager can be used incorrectly, it is recommended that customers incorporate the use of Configuration Transport Manager into their own release process to give due attention to the contents of a change set, and to ensure the correct and timely activation of a change set on the production system.

For more information about the recommended procedures for managing the creation and modification of business objects, see Defining a data management policy on page 11.

Reverting to a defined set of configuration data

Making configuration changes on a production environment is often considered a high risk activity. The Configuration Transport Manager provides undo functionality to mitigate against errors encountered after applying the configuration data in the transported change set. Undo functionality allows a customer to revert the system back to the configuration before the change set was applied.

Undo option limitation

The undo option is useful only for a limited period of time. Once live runtime data is created based on the newly transported configuration data, the revert feature can no longer be used as the configuration data and runtime data are out of sync. When the application starts to use the configuration data, runtime data is created based on the latest set of configuration data. To revert to an earlier configuration at this stage also requires reverting the runtime data. To do this, a database rollback is necessary.

> **Note:** When logically deletable business objects are reverted, they are logically deleted and not physically deleted. The system state when a logically deleted business object is reverted is not exactly the same as it was before the business object was applied. There are now logically deleted records on the database that were not there previously.

The revert change set

At the time of applying a change set, a revert change set is automatically created by the system. This revert change set contains a snapshot of the business object before applying the change set. After a user applies a change set, they can select the undo action for the change set, which reverts the system back to the original configuration by applying the contents of revert change set. After reverting, the system is returned to the state it was in before the transported change set was applied. This is done by deleting the business objects that were newly added to the target system. The business objects that were updated because of the transported change set are reverted back to the previous state by using the business objects in the automatically created change set.

Accountability and the change set history

The Configuration Transport Manager provides a number of features to support accountability and traceability. History information such as when a change set was transported or applied is also maintained to provide an audit trail.

The Configuration Transport Manager provides accountability with respect to the various operations performed by different users in transporting business objects from one system to another. This helps to track operations performed by multiple users and allows remedial actions to be taken if the system goes into an unstable or unpredictable state through the use of the Configuration Transport Manager. To support tracking the change set, the name of the change set must always be unique. The contents of a change set are frozen before it can be moved across systems so that the contents of an applied change set are always known.

Change set transport history

Each time a released change set is transported, both automatically and manually, an entry is recorded in the transport history for the change set. Both inbound and outbound transport history is maintained on each system to facilitate the full audit trail of a change set.

For example in a three system landscape for development, testing, and production, the following entries are created when transporting the 'Release 0618' change set. When the change set is transported from development to test, the outbound transport history entry is captured on the development system and the inbound transport history is captured on the testing system. When the change set is transported from testing to production, the outbound transport history entry is captured on the testing system and the inbound transport history is captured on the production system.

The transport history details include the transport type, automatic or manual, the target system to which the change set was transported, the user who transported the change set, and the date and time when the change set was transported.

Change set apply history

Every time a released change set is applied on a system an entry is recorded in the apply history for the change set on the system in question. To get a complete history of when a change set was applied, a user can look at the history across the systems in the landscape.

In the example used previously, where the 'Release 0618' change set is transported from development to testing to production, when the change set is transported and applied on the testing system, an entry is created on the testing system in the apply history for the 'Release 0618' change set. When the change set is transported and applied on the production system, an entry is created on the production system in the apply history for the 'Release 0618' change set.

The apply history details include the user who applied the change set and the date and time when the change set was applied.

Change set undo history

Every time a released change set is applied on a system an entry is recorded in the apply history for the change set on the system in question. To get a complete history of when a change set was applied, a user would look at the history across the systems in the landscape.

In the example used previously where the "Release 0618" change set is transported from development to testing to production: When the change set is transported and applied on the testing system, an entry is created on the testing system in the apply history for the "Release 0618" change set. When the change set is transported and applied on the production system, an entry is created on the production system in the apply history for the "Release 0618" change set.

The apply history details include the user who applied the change set and the date and time with the change set was applied.

1.4 Physical table to administration business object mapping

Use this information to understand the process of physical table to administration business object mapping in the Configuration Transport Manager. For each business object, a description is provided and a list of the entities that are impacted by transporting an instance of that business object. Any special conditions that are associated with the business object are also described.

> **Note:** The business objects listed are available only in the Configuration Transport Manager if a suitable license is in place for the area in question.

Merative ™ SPM Platform

A list of the business objects that contain configuration data that relate to Merative ™ SPM Platform and the entities that store the data. These include business objects that relate to the following areas: Case, Cúram Express Rules, Organization, Case Audits, Workflow, and Cúram Intelligent Evidence Gathering.

Case

A list of the business objects that contain configuration data that relate to cases and the entities that store the data.

Benefit Product

The Benefit Product business object contains the configuration data for the product delivery case of type benefit product. The benefit product case type determines whether a person or employer meets the eligibility criteria for a benefit.

A successful eligibility outcome usually results in the benefit being delivered in the form of a payment. The following entities store the configuration data that is contained within the business object.

- PRODUCT
- PRODUCTCATEGORY
- PRODUCTCONCERNROLE
- PRODUCTAPPEALPROCESS
- APPEALSTAGE
- EVIDENCEMETADATA
- PRODUCTEVIDENCELINK
- TEMPORALEVIDENCEAPPROVALCHECK
- EVIDENCESCREEN
- EVIDENCEAPPROVALCHECK
- ADHOCBONUSCRITERIA
- PDASSESSMENTCONFIGURATION
- CASEAPPROVALCHECK
- DEDUCTIONPRODUCTLINK
- DEDUCTION
- PRODUCTDELIVERYPATTERNINFO
- PRODUCTDELIVERYPATTERN
- PRODUCTPROVISION
- PRODUCTRULESLINK
- PRODUCTTIMECONSTRAINT
- DELIVERYMETHOD
- PROVISIONLOCATION
- MILESTONELINK
- MILESTONECONFIGURATION
- LOCALIZABLETEXT

- TEXTTRANSLATION
- SECURITYIDENTIFIER

Integrated Case

The Integrated Case business object contains the configuration data for an integrated case type. An integrated case is used to manage and consolidate case information for one or more members of the same household or family unit.

The following entities store the configuration data that is contained within the business object.

- `ADMININTEGRATEDCASE`
- `ICASSESSMENTCONFIGURATION`
- `EVIDENCEMETADATA`
- `ADMINICEVIDENCELINK`
- `MILESTONECONFIGURATION`
- `MILESTONELINK`
- `LOCALIZABLETEXT`
- `TEXTTRANSLATION`

Liability Product

The Liability Product business object contains the configuration data for the product delivery case of type liability product. Liabilities are delivered to persons or employers in the form of a bill. For example, a person may be billed for an overpaid benefit.

The following entities store the configuration data that is contained within the business object.

- PRODUCT
- PRODUCTCATEGORY
- PRODUCTCONCERNROLE
- EVIDENCEMETADATA
- PRODUCTEVIDENCELINK
- TEMPORALEVIDENCEAPPROVALCHECK
- EVIDENCESCREEN
- EVIDENCEAPPROVALCHECK
- CASEAPPROVALCHECK
- PRODUCTDELIVERYPATTERNINFO
- PRODUCTDELIVERYPATTERN
- PROVISIONLOCATION
- PRODUCTPROVISION
- PRODUCTRULESLINK
- PRODUCTTIMECONSTRAINT
- SECURITYIDENTIFIER
- LOCALIZABLETEXT
- TEXTTRANSLATION

Assessment

The Assessment business object contains the configuration data for an assessment case type. The purpose of the assessment case type is to perform an initial evaluation to determine whether a person or prospect person potentially is eligible for a benefit. Each assessment case represents a program for which the organization wants to assess potential eligibility.

The following entities store the configuration data that is contained within the business object.

- `ASSESSMENTCONFIGURATION`
- `ASSESSMENTRULESLINK`

CER Benefit Product

The CER Benefit Product business object contains the configuration data for the product delivery case of type Cúram Express Rules (CER) Benefit Product. The CER Benefit Product case type determines whether a person or employer meets the eligibility criteria for a benefit.

A successful eligibility outcome usually results in the benefit delivered in the form of a payment. CER Benefit Products are configured to use Cúram Express Rules.

> **Note:** The following special behaviors apply to a CER Benefit Product business object.
>
> 1. The user must manually publish the CER Rule Data Configuration on the target system.
> 2. When using CER rule sets, certain changes to the system can cause widespread recalculations, such as case reassessments to occur. As a result, when transporting a mix of object types, where some require publishing, the user must publish the following object types in this order: Rate Table, CER Data Configuration, CER Rule Set and CER Benefit Product.
> 3. If transporting a new CER Benefit Product that does not exist on the target system, the user must transport the Product Name code table as a dependent business object.

The following entities store the configuration data that is contained within the business object.

- PRODUCT
- CREOLEPRODUCT
- CREOLEPRODUCTSANDBOX
- PRODUCTCATEGORY
- PRODUCTCONCERNROLE
- PRODUCTEVIDENCELINK
- EVIDENCEMETADATA
- TEMPORALEVIDENCEAPPROVALCHECK
- ADHOCBONUSCRITERIA
- PRODUCTPROVISION
- PROVISIONLOCATION
- DEDUCTIONPRODUCTLINK
- DEDUCTION
- CASEAPPROVALCHECK
- EVIDENCEAPPROVALCHECK
- PRODUCTTIMECONSTRAINT
- PDASSEMENTCONFIGURATION
- EVIDENCESCREEN
- PRODUCTDELIVERYPATTERN
- PRODUCTDELIVERYPATTERNINFO
- DELIVERYMETHOD
- MILESTONELINK
- MILESTONECONFIGURATION
- ABSENCEREASONCONFIGURATION

- LOCALIZABLETEXT
- TEXTTRANSLATION
- PRODUCTAPPEALPROCESS
- APPEALSTAGE

CER Liability Product

The CER Liability Product business object contains the configuration data for the product delivery case of type CER Liability Product. The CER Liability Product case type determines whether a person or employer meets the liability criteria for the product.

A successful liability outcome usually results in the payments being delivered to the organization. CER Liability Products are configured to use Cúram Express Rules.

> **Note:** The following special behaviors apply to a CER Liability Product business object.
> 1. The user must manually publish the CER Rule Data Configuration on the target system.
> 2. If transporting a new CER Liability Product that does not exist on the target system, the user must transport the Product Name and Product Type code tables as a dependent business object.

The following entities store the configuration data that is contained within the business object.

- PRODUCT
- CREOLEPRODUCT
- CREOLEPRODUCTSANDBOX
- PRODUCTCATEGORY
- PRODUCTCONCERNROLE
- PRODUCTEVIDENCELINK
- EVIDENCEMETADATA
- TEMPORALEVIDENCEAPPROVALCHECK
- PRODUCTPROVISION
- PROVISIONLOCATION
- PROVIDERLOCATION
- CASEAPPROVALCHECK
- EVIDENCEAPPROVALCHECK
- PRODUCTTIMECONSTRAINT
- EVIDENCESCREEN
- PRODUCTDELIVERYPATTERN
- PRODUCTDELIVERYPATTERNINFO
- DELIVERYMETHOD
- MILESTONELINK
- MILESTONECONFIGURATION
- LOCALIZABLETEXT
- TEXTTRANSLATION

Correction Product

The Correction Product business object contains the configurations data for a payment correction product. The purpose of a payment correction is to support the creation of granular overpayment and underpayment instruction line items.

The following entities store the configuration data that is contained within the business object.

- PRODUCT
- PRODUCTCATEGORY
- PRODUCTCONCERNROLE
- PRODUCTPROVISION
- PROVISIONLOCATION
- DEDUCTIONPRODUCTLINK
- DEDUCTION
- CASEAPPROVALCHECK
- PRODUCTTIMECONSTRAINT
- PRODUCTDELIVERYPATTERN
- PRODUCTDELIVERYPATTERNINFO
- DELIVERYMETHOD
- MILESTONELINK
- MILESTONECONFIGURATION
- LOCALIZABLETEXT
- TEXTTRANSLATION

Investigation

The Investigation business object contains the configuration data for an investigation case type. Investigation cases are used to inquire into circumstances that surround an allegation or incident.

Organizations receive thousands of reports each year that must be investigated. Examples of types of allegations that require investigation include child maltreatment or benefit fraud. The following entities store the configuration data that is contained within the business object.

- INVESTIGATIONCONFIG
- INVCASSESSMENTCONFIG
- INVESTIGATIONAPPROVALCHECK
- MILESTONECONFIGURATION
- MILESTONELINK
- RESOLUTIONCONFIGURATION
- INVESTIGATIONRESOLUTIONLINK
- LOCALIZABLETEXT
- TEXTTRANSLATION
- SECURITYIDENTIFIER
- SOCIALENTERPRISEFOLDERADMIN
- CASECOLLABORATIONADMINLINK

Issues

The Issues business object contains the configuration data for an issue. Issues are created to manage case-related problems or questions identified by caseworkers during the course of their work.

Issues also can be created to manage evidence discrepancies identified during rules processing. An issue can be created manually by a user or automatically by the system. The following entities store the configuration data that is contained within this business object.

- `ISSUECONFIGURATION`
- `APPEALPROCESS`
- `APPEALSTAGECONFIGURATION`
- `ISSUERESOLUTIONLINK`
- `ISSUERESOLUTIONAPPROVALCHECK`
- `ISSUETIMECONSTRAINT`
- `RESOLUTIONCONFIGURATION`

Screening

The Screening business object contains the configuration data for a screening case type. A screening case type is used to handle a number of specific program assessments which are run to determine which members of a household are potentially eligible for a range of benefits.

A screening case type acts as a container for consolidating information. The following entities store the configuration data that is contained within the business object.

- SCREENINGCONFIGURATION
- SCREENINGASSESSMENTCONFIG
- MILESTONECONFIGURATION
- MILESTONELINK
- LOCALIZABLETEXT
- TEXTTRANSLATION

Cúram Express Rules

A list of the business objects that contain configuration data that relate to Cúram Express Rules (CER) and the entities that store the data.

CER Rule Set

The CER Rule Set business object contains the configuration data for the CER rule set. It contains the CER rule set and any category links for that rule set.

The CER rule set business object is dependent on the CER rule set category business object.

> **Note:** The following special behaviors apply to the CER Rule Set business object.
>
> 1. The user must manually publish the CER rule sets on the target system.
> 2. When using CER rule sets, certain changes to the system can cause widespread recalculations, such as case reassessments, to occur. This is due to the way in which data is fed into, and stored for CER rule sets, and how this gets reorganized to ensure it always reflects the latest data and configurations in the system. As a result, when transporting a mix of object types, where some require publishing, publish the following object types in this order: Rate Table, CER Data Configuration, CER Rule Set and CER Benefit Product.
> 3. The revert functionality for a CER rule set varies depending on whether or not the user has published the CER rule set on the target system. If the transported CER rule set has not yet been published, then the revert functionality will remove the unpublished CREOLERuleSetEditAction record, and will also remove any associated rule set category link records. If the transported CER Rule Set has been published, then a CREOLERuleSetEditAction record of type "Remove" will be written to the target system for the CER rule set, and any associated rule set category link records that exist will be removed. The CREOLERuleSetEditAction must then be published to fully remove the CER rule set from the target system.

The following entities store the configuration data that is contained within the business object.

- CREOLERULESETEDITACTION
- CREOLERULESETCATEGORYLINK

CER Rule Object Propagator

The CER Rule Object Propagator business object contains the configuration data for the CER rule object propagator. It contains the CER data configuration details.

The business object contains the configuration data for the CER rule object propagator. It contains the CER data configuration details.

> **Note:** The following special behaviors apply to a CER rule object propagator business object:
>
> 1. The user must manually publish the CER rule sets on the target system.
> 2. When using CER rule sets, certain changes to the system can cause widespread recalculations, such as case reassessment, to occur. This is due to the way in which data is fed into, and stored for CER rule sets, and how this gets reorganized to ensure it always reflects the latest data and configurations in the system. As a result, when transporting a mix of object types, where some require publishing, publish the following object types in this order: Rate Table, CER Data Configuration, CER Rule Set and CER Benefit Product.
> 3. The revert functionality for a CER rule set varies depending on whether or not the user has published the CER rule set on the target system. If the transported CER rule set has not yet been published, then the revert functionality will remove the unpublished CREOLERuleSetEditAction record, and will also remove any associated rule set category link records. If the transported CER Rule Set has been published, then a CREOLERuleSetEditAction record of type Remove is written to the target system for the CER rule set, and any associated rule set category link records that exist are removed. The CREOLERuleSetEditAction must be published to fully remove the CER rule set from the target system.

The following entities store the configuration data that is contained within the business object.

- RULEOBJPROPCONFIGSANDBOX
- LOCALIZABLETEXT
- TEXTTRANSLATION

CER Rule Set Category

The CER Rule Set Category business object contains the configuration data for the CER rule set category.

The CER rule set category describes and categorizes CER Rule Sets. The following entities store the configuration data that is contained within the business object.

- CREOLERULESETCATEGORY
- LOCALIZABLETEXT
- TEXTTRANSLATION

Organization

A list of the business objects that contain configuration data that relate to the organization and the entities that store the data. The Organization Structure, Location Structure, User, and External User business objects, and their associated entities, are described.

Table 2: Organization business objects and entities

Business object	Description	Entities
Organization Structure	Contains the configuration data for an organization structure. The organization structure provides a way to manage the organization's users and functional areas. The organization structure allows for the partition of administration functionality that relates to an organization's hierarchy structure across many lines.	- ORGANISATIONSTRUCTURE - ORGANISATION - ORGANISATIONUNIT - ACTIVITY - ACTIVITYRECURRENCE - ORGUNITPARENTLINK - ORGUNITPOSITIONLINK - POSITION - JOB - ORGANISATIONUNITRESOURCE - RESOURCES - POSITIONREPORTINGLINK - POSITIONLOCATIONLINK - CASEAPPROVALCHECK - TEMPORALEVIDENCEAPPROVALCHECK - INVESTIGATIONAPPROVALCHECK - ISSUERESOLUTIONAPPROVALCHECK - MILESTONEWAIVERAPPROVALCHECK - EVIDENCEAPPROVALCHECK - SECURITYIDENTIFIER - LOCALIZABLETEXT - TEXTTRANSLATION

Business object	Description	Entities
Location Structure	Contains the configuration data for a location structure. The location structure maintains the locations of an organization and the parent-child relationship between the locations.	• LOCATIONSTRUCTURE • LOCATION • WORKINGPATTERN • ADDRESS • LOCATIONACCESS • LOCATIONHOLIDAYLINK • LOCATIONHOLIDAY • LOCATIONRESOURCELINK • RESOURCES • DAILYSCHEDULE • SLOT • EMAILADDRESS • POSITIONSLOTAVAILABILITY • PHONENUMBER • SECURITYIDENTIFIER
User	Contains the configuration data for a user. Users are registered on the system and the user information includes contact details, positions and security access rights.	• USERS • PHONENUMBER • EMAILADDRESS • USERSKILL • RESOURCES • ACTIVITY • ACTIVITYRECURRENCE • TASKREDIRECTION • CASEAPPROVALCHECK • EVIDENCEAPPROVALCHECK • SERVICEPLANAPPROVALCHECK • TEMPORALEVIDENCEAPPROVALCHECK • INVESTIGATIONAPPROVALCHECK • ISSUERESOLUTIONAPPROVALCHECK • MILESTONEWAIVERAPPROVALCHECK • WORKINGPATTERN • POSITIONHOLDERLINK
External User	Contains the configuration data for an external user. An external user is a user of the system who is not part of the organization structure. They access the application over the web and have restricted access to the system. External User accounts are managed separately to internal user accounts.	• EXTERNALUSER • EXTERNALUSERPARTICIPANTLINK

Dynamic Features

A list of the business objects that contain configuration data that relate to dynamic features and the entities that store the data. The Dynamic Evidence Type, Dynamic Evidence Type Version, Dynamic Product Evidence Type, and Integrated Case Evidence Type are described.

Dynamic Evidence Type

The business object contains the configuration data for dynamic evidence types. Dynamic evidence types are the administrative equivalent of non-dynamic evidence types – a logical grouping of related attributes about which an organization wants to record information in respect

of a case, for example, Income Evidence. The following entities store the configuration data that is contained within the business object.

- EVIDENCETYPEDEF
- EVIDENCETYPEDEFINITION

Dynamic Evidence Type Version

The business object contains the configuration data for dynamic evidence type versions. Dynamic evidence types can have multiple versions which vary over time. Dynamic evidence allows for information recorded in respect of 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 remains effective until the next version.

- EVIDENCETYPEVERSIONDEF
- EVIDENCERULESETDEF

Dynamic Product Evidence Type

The business object contains the configuration data for dynamic product evidence type links. Products are linked to dynamic evidence types through product evidence type links. This business object allows an administrator to add product evidence type link data to a change set to apply to a target system. During the apply changes process in the target system, the dependent business objects are first created if they do not exist, and then the product evidence type links are created.

> **Note:** Transported links between products and dynamic evidence types are not visible on the target system unless at least one dynamic evidence type version in an Active status exists on the dynamic evidence type. However, when dynamic evidence type versions are transported, their status in the target system is InEdit. Therefore, before using this business object for a new product and evidence type, the administrator must first transport the relevant dynamic evidence type version business object, then activate the transported dynamic evidence type versions, and transport the relevant link business object.

The following entities store the configuration data that is contained within the business object.

- PRODUCTEVIDENCETYPEDEFLINK
- PRODUCT
- EVIDENCETYPEDEF

Integrated Case Evidence Type

The business object contains the configuration data for integrated case evidence type links. Integrated cases are linked to dynamic evidence types through product evidence type links. This business object allows an administrator to add integrated case evidence type link data to a change set to apply to a target system. During the apply changes process in a target system, the dependent business objects are first created if they do not exist, and then the integrated case evidence type links are created.

> **Note:** Transported links between integrated cases and dynamic evidence types are not visible on the target system unless at least one dynamic evidence type version in an Active status exists on the dynamic evidence type. However, when dynamic evidence type versions are transported, their status is InEdit. Therefore, before using this business object for a new integrated case and evidence type, the administrator must first transport the relevant dynamic evidence type version business object, then activate the transported dynamic evidence type versions, and transport the relevant link business object.

The following entities store the configuration data that is contained within the business object.

- ADMINICEVIDENCETYPEDEFLINK
- ADMININTEGRATEDCASE
- EVIDENCETYPEDEF

Case Audits

A list of the business objects that contain configuration data that relate to case audits and the entities that store the data. The Case Audit Configuration and Case Audit Selection Query business objects are described.

Table 3: Case audit business objects and entities

Business object	Description	Entities
Case Audit Configuration	Contains the configuration data for an audit of a particular case type. A case audit is an examination or evaluation of a case or set of cases. The purpose of a case audit is to monitor the delivery of benefits and services to clients. A range of information on quality, practice, and outcomes can also be gathered and assessed. **Note:** Case audit configurations cannot be removed as every case type must have a configuration.	• AUDITCASECONFIG • AUDITCASEFOCUSAREA • AUDITCASESELECTIONQUERY
Case Audit Selection Query	A selection query is used to generate a sample of cases and contain the selection criteria that are used to search for and produce the list of cases. **Note:** Active selection queries cannot be removed.	• SELECTION QUERY • CRITERIA • SELECTIONQUERYCRITERIALINK

Business Intelligence and Reporting Tools

A list of the business objects that contain configuration data that relate to the Business Intelligence and Reporting Tools® (BIRT) and the entities that store the data. The Report Configuration and Viewer Configuration business objects are described.

Table 4: Business Intelligence and Reporting Tools® business objects and entities

Business object	Description	Entities
Report Configuration	Contains the configuration for displaying a report. Multiple configurations can be created for each report using different logical names. This means that the same report to be re-used with different display options, depending on the requirements. BI report configuration allows the user to define the report display details and parameters. **Note:** When transporting BI viewer configuration details, its root parameter is not transported as the target system likely has a different root. The system administrator must set the root parameter on the target system after transporting.	• BIREPORTCONFIGURATION • BIREPORTPARAMETER
Viewer Configuration	Contains the configuration data for the BI viewer details. This allows the user to change from the default viewer, the BIRT viewer, to use the Actuate® BIRT viewer. **Note:** For transported BI report configuration to be functional the target system must have BI viewer configuration set up.	• BIVIEWERCONFIGURATION • BIREPORTPARAMETER

Workflow

A list of the business objects that contain configuration data that relate to workflow and the entities that store the data. The Work queue and Allocation Target business objects are described.

Table 5: Workflow business objects and entities

Business object	Description	Entities
Workflow	Contains the configuration data for a work queue. A work queue is a subscription based list that can be assigned as a target item for an allocation target. They are maintained in Workflow Administration where an administrator can create, edit or remove work queues from the system. An administrator can subscribe users to the work queue or the users themselves can subscribe to the work queue.	• WORKQUEUE • WORKQUEUESUBSCRIPTION

Business object	Description	Entities
Allocation Target	Contains the configuration data for an allocation target. Allocation targets are logical grouping of users or work queues to which tasks and notifications can be assigned. This allows tasks and notifications to be assigned to a cross section of users in the organization in a manner that can be customized by a workflow administrator.	• ALLOCATIONTARGET • ALLOCATIONTARGETITEM

Miscellaneous

A list of miscellaneous business objects that contain configuration data and the entities that store the data. The Rate Table, Decision Assist assessment type, Financial Calendar, Language and Locale Mapping, Intake Process, Priority, Complexity, Risk, and Advisor business objects are described.

Rate Table

The Rate Table business object contains the configuration data for a rate table. Rates are values that can vary over time. An example of a rate is the gross income and net income limits based on the number of people in a household. Rate tables are used to maintain these rates.

> **Note:** When using CER rule sets, certain changes to the system can cause widespread recalculations, such as case reassessments to occur. This is due to how the data is reorganized to ensure that it always reflects the latest data and configurations in the system. As a result, when transporting a mix of object types, where some require publishing, it recommended that you publish the following business object types in this order: Rate Table, CER Data Configuration, CER Rule Set, and CER Benefit Product.

The following entities store the configuration data that is contained within the business object.

- RATEHEADER
- RATEROW
- RATECOLUMN
- RATECELL

Financial Calendar

The Financial Calendar business object contains the configuration data for the financial calendar. The financial calendar is used to reflect when the office is open or closed and to display processing or non-processing dates.

The following entity stores the configuration data that is contained within the business object.

- FINANCIALCALENDAR

Language and Locale Mapping

The Language and Locale business object contains the configuration data for the language and locale mapping. Language and locale mappings are used to customize the user interface language. They are critical to many culturally and linguistically sensitive data operations, for example, locale information is used when generating pro forma communications. Each language has a single locale associated with it. For example, English is mapped to the en locale.

The following entity stores the configuration data that is contained within the business object.

- FINANCIALCALENDAR

Intake Process

Common Intake (CCI) enables an agency to define the intake process to meet its requirements by providing configuration settings for several elements of CCI including triage, screening, application, life event, referral, and Priority Complexity Risk (PCR) processing. The agency is able to configure functions by user role.

The following entity stores the configuration data that is contained within the business object.

- `INTAKEPROCESSNAME`
- `INTAKEPROCESSCONFIGURATION`
- `INTAKEPROCESSROLEASSIGNMENT`
- `MILESTONECONFIGURATION`
- `PROGRAMTYPE`

Priority, Complexity, Risk

The business object contains the data for a Priority Complexity Risk (PCR) configuration. PCR is a mechanism which enables agencies to provide a differential response when dealing with a client. Priority is the time line in which a case or application must be dealt with. Complexity refers to the complexity of the situation, based on, for example, the number of household members. Risk delineates the potential impact to the agency, for example, a potentially long term claim could be considered high risk to the agency.

The following entities store the configuration data that is contained within the business object.

- PCRCONFIGURATION
- PCRRANGEMAPPING

Advisor

The business object contains the data for an advisor configuration. The Advisor is a dynamic caseworker tool that provides context sensitive tips and reminders to the worker throughout the intake process. It is built upon the Cúram Express Rules (CER) Engine. The Advisor analyzes data that is entered (or known if the client was previously registered on the system) and guides the worker towards areas that may require his or her attention.

The following entities store the configuration data that is contained within the business object.

- ADVICECONTEXTCONFIGURATION
- ADVICECONTEXTRULES
- ADVICECONTEXTKEY
- CONFIGURATIONPARAMETER

Decision Assist Assessment Type

The Decision Assist Assessment Type business object contains the configuration data for an assessment case of type decision assist. A decision assist assessment contains the questionnaires and rules that are used when running assessments for clients using Cúram Decision Assist (CDA), to calculate an outcome or rating for a client.

The following entities store the configuration data that is contained within the business object.

- DETERMINATIONCONFIG
- DETASSESSMENTCONFIGLINK
- DETERMINATIONPACKAGEVERSION
- DETERMINATIONPACKAGE
- DPQUESTIONNAIRELINK
- DPDECISIONMATRIXLINK
- DATRANSLATION
- QUESTIONNAIREVERSION
- DECISIONMATRIX
- DECISIONMATRIXVERSION
- DAOUTCOME
- DMOUTCOMELINK
- DMQUESTIONLINK
- OUTCOMESCOREASSOCIATION
- QUESTIONNAIRE
- TPQUESTIONNAIRELINK
- QUESTION
- QUESTIONALIAS
- CONDITIONDCLINK
- QUESTIONOUTCOMEMETADATA
- CONDITION
- ICDCODECONDITIONLINK
- ICDCODEVERSION
- ICDCODE
- TYPICALPICTURE
- TYPICALPICTUREANSWER
- TYPICALPICTUREVERSION
- TPDECISIONOUTCOME
- ASSESSMENT

Service Plan

The Service Plan business object contains the configuration data for a service plan. Service Planning is a mechanism by which a social enterprise organization can plan, in an integrated manner, the delivery of services and activities to help a client achieve a specific goal.

The type of service plan created by an organization depends upon the specific goal that the plan is intended to achieve. Common service plan types include Return to Work, Child Protection, and Permanency Plans.

The following entities store the configuration data that is contained within the business object.

- SERVICEPLAN
- SERVICEPLANAPPROVALCHECK
- GOAL
- GOALCONTRACTTEXT
- SERVICEPLANGOALLINK
- ADMINICSERVICEPLANLINK
- MILESTONECONFIGURATION

- MILESTONELINK
- PLANTEMPLATESERVICEPLANLINK
- PLANITEM
- OUTCOME
- GOALOUTCOMELINK
- GOODCAUSE
- CONTRACTTEXT
- PLANITEMCONTRACTTEXT
- PLANITEMOUTCOMELINK
- PLANITEMGOODCAUSELINK
- PLANITEMAPPROVALCRITERIALINK
- APPROVALCRITERIA
- SUBGOAL
- GOALSUBGOALLINK
- SUBGOALCONTRACTTEXT
- SUBGOALPLANITEMLINK
- PLANTEMPLATE
- PLANTEMPLATESUBGOAL
- PLANTEMPLATEPLANITEM
- PLANTEMPLATEPLANITEMAPPRCRIT
- PLANTEMPLATEMILESTONE
- SERVICEPLANGROUP
- SERVICEPLANGROUPLINK
- PLANTEMPLATEPLANGROUP
- SUBGOALOUTCOMELINK
- TASKCONFIGURATION
- SECURITYIDENTIFIER
- LOCALIZABLETEXT
- TEXTTRANSLATION

Cúram Intelligent Evidence Gathering

A list of the business objects that contain configuration data that relate to workflow and the entities that store the data. The Intelligent Evidence Gathering Scripts and Datastore Schema business objects are described.

Table 6: Cúram Intelligent Evidence Gathering business objects and entities

Business object	Description	Entities
Intelligent Evidence Gathering	Contains the configuration data for a Cúram Intelligent Evidence Gathering (IEG) script. With IEG, information is gathered interactively by displaying a script of questions that a user can provide answers to. Questions are only displayed if they are consistent with the user's previous answers so that the user is only required to provide answers relevant to his or her needs and situation.	• IEGSCRIPTINFO • IEGSCRIPTRELS • APPRESOURCE

Business object	Description	Entities
Datastore Schema	Contains the data for a datastore schema. Datastore schemas define the structure of entities that can be stored in the application datastore. The configuration data that is contained within this business object is stored on the following entity:	• DATASTORESCHEMA

Cúram Funded Program Management

A list of the business objects that contain configuration data that relate to Cúram Funded Program Management and the entities that store the data. The Fund and FundingRuleSet business objects are described.

Table 7: Cúram Funded Program Management business objects and entities

Business object	Description	Entities
Fund	Contains the configuration data for a fund. A fund can have an administrative association to a product or service, and is segmented into a number of fiscal years, each of which can contain its own budget. This allows an agency to define how a fund will be structured over time, and when it will be obligated.	• PROGRAMFUND • FUNDCONFIGURATION • FISCALYEAR • FUNDFISCALYEAR • FUNDUSERROLE • ORGOBJECTLINK
FundingRuleSet	Contains the configuration data for a funding rule set. A product or service can have more than one related fund. In this case, rules are required to define the order to use the rules.	• FUNDEDITEMRULESLINK

Infrastructure

A list of the business objects that contain configuration data that relates to infrastructure and the entities that store the data. They include the following business objects: Rule Set, Application Property, Code Table, Workflow, Event, Application Resource, Extensible Stylesheet Language Template, Microsoft® Word Template, Security, and User Interface.

> **Note:** Some infrastructure business objects exhibit special behavior. A target system is created by using transported data from the source system. In the source system, a primary key can be updated. If the primary key is then transported, it is added as a new business object rather than an update to the existing one on the target system. Both the original primary key name and the updated primary key are created in the target system. The business objects that are affected by this behavior are Security Role, Security Group, Security Identifiers, Code Table, and Properties.

Table 8: Infrastructure business objects and entities

Business object	Description	Entities
Rule Set	Contains the configuration data for a rule set. A rule set is the set of rules that is applied to case evidence to determine eligibility for a product. At least one rule set must be assigned to a product as a prerequisite to eligibility determination.	• RULESETINFORMATION • RULESETLINK
Application Property	Contains the configuration data for an application property. The properties table contains all the properties that are used when an application is run. The properties are loaded to the database during the build database target and at run time are cached from the database for use by the Application. The properties that are defined can be dynamic or static.	• PROPERTIES • PROPDESCRIPTION
Code Table	Contains the configuration data for a code table. Code tables contain codes for items that appear in the drop down fields. A code table is made up of a number of code table items; each code table item represents a selection in a drop down field.	• CODETABLEHEADER • CODETABLEITEM
Code Table Hierarchy	Contains the configuration data for a code table hierarchy record. A code table hierarchy is where multiple code tables are linked into a number of levels. Any number of code tables can be included in a code table hierarchy. A code table hierarchy allows the values available for selection in the drop-down field for one code table to be determined by the value selected in the drop-down field for another code table.	• CODETABLEHIERARCHY
Workflow Business Process	Contains the configuration data for a workflow process definition. A workflow process is enacted at run time based on a workflow process definition. A workflow process definition describes the flow of a business process in terms that are understood by the Workflow Engine. It defines the activities in the workflow and the transitions between them. It also defines the data that is passed into and out of an activity during the execution of a workflow process instance.	• PROCESSDEFINITION
Workflow Process Enactment	Contains the configuration data for a workflow process enactment event configuration. Events provide a mechanism for loosely coupled parts of the application to communicate information about state changes in the system. A process is started in response to the raising of an event and requires the setup of workflow process enactment event configuration data. The configuration specifies the process or processes to start in response to the raising of a specific event. Process enactment event configuration is stored on the database and a user interface is supplied to allow the manipulation of this data.	• PROCENACTMENTEVT • PROCENACTEVTDATA

Business object	Description	Entities
Event	Contains the configuration data for an event. Events provide a mechanism for loosely coupled parts of the application to communicate information about state changes in the system. When one module in the application raises an event, one or more other modules receive a notification about the occurrence of the event provided they are registered as listeners for that event.	• EVENTCLASS • EVENTTYPE
Microsoft® Word template.	Contains the configuration data for a Microsoft® Word template. A user can create a communication based on a Microsoft® Word template and modify the communication, as necessary before the user issues the communication.	• DOCUMENTTEMPLATE
Extensible Stylesheet Language Template	contains the configuration data for a Extensible Stylesheet Language (XSL) template. XSL templates are used to automatically generate bulk communications, often through batch processing. An XSL template includes standard text that is displayed on all communications that are generated from the template.	• XSLTEMPLATE • XSLTEMPLATEINST

Application Resource

A list of the business objects that related to the application resource and the entities that store the data.

Table 9: Application resource business objects and entities

Business object	Description	Entities
Application Resource Universal Access	Contains the data for a Universal Access (UA) stylesheet application resource. It can be used by the system in a variety of ways; some alter the functionality provided by the system and therefore they configure the system to suit an organization's needs.	• APPRESOURCE
Application Resource Image	Contains the data for an image application resource. It can be used by the system in a variety of ways; some alter the functionality provided by the system and therefore they configure the system to suit an organization's needs	• APPRESOURCE
Application Resource Properties	Contains the data for a properties application resource. It can be used by the system in a variety of ways; some alter the functionality provided by the system and therefore they configure the system to suit an organization's needs.	• APPRESOURCE
Application Resource Schema	Contains the data for an XML Schema Definition (XSD) schema application resource. It can be used by the system in a variety of ways; some alter the functionality provided by the system and therefore they configure the system to suit an organization's needs.	• APPRESOURCE

Business object	Description	Entities
Application Resource XML	Contains the data for an XML application resource. It can be used by the system in a variety of ways; some alter the functionality provided by the system and therefore they configure the system to suit an organization's needs.	• APPRESOURCE

Security

A list of the business objects that contain configuration data that relates to security and the entities that store the data. These include the following business objects: Security Role, Security Group, and Security Identifier.

Table 10: Security business objects and entities

Business object	Description	Entities
Security Role	Contains the configuration data for a security role. A security role is a configuration used to manage one or more security groups associated to a user.	• SECURITYROLE • SECURITYROLEGROUP
Security Group	Contains the configuration data for a security group. Security group is a configuration used to manage one or more security identifiers associated to a user.	• SECURITYGROUP • SECURITYGROUPSID
Security Identifier	Contains the configuration data for security identifiers. Security identifier is a configuration used to manage one or more securable elements of the application associated to a user, for example a method or a field.	• SECURITYIDENTIFIER • SECURITYFIDSID
Application Resource Schema	Contains the data for an XML Schema Definition (XSD) schema application resource. It can be used by the system in a variety of ways; some alter the functionality provided by the system and therefore they configure the system to suit an organization's needs.	• APPRESOURCE
Application Resource XML	Contains the data for an XML application resource. It can be used by the system in a variety of ways; some alter the functionality provided by the system and therefore they configure the system to suit an organization's needs.	• APPRESOURCE

User Interface

A list of the business objects that related to the user interface and the entities that store the data.

Table 11: User interface business objects and entities

Business object	Description	Entities
User Interface Resource	Contains the configuration data for a user interface application. This application is a collection of user interface components combined to create specific content for a particular user or role. For example, an Administration application defines the application for a user with an administrative role. T	• APPRESOURCE

Business object	Description	Entities
User Interface Section Configuration	Contains the configuration data for a user interface section. An application can contain a number of sections. The section allows quick and easy access to the more common tasks for a user. For example, a section can be created to allow easy access to the users Inbox. The section can also contain a shortcut panel which provides quick links to content relevant for the section.	• APPRESOURCE
User Interface Tab Configuration	Contains the configuration data for a user interface tab. A tab represents a business object or logical grouping of information. A section can contain one or more tabs, each of which displays content relevant to the section.	• APPRESOURCE
User Interface Quick Links Configuration	Quick links are created for the organization and can be associated with a user's security role. When a user logs into the system, any quick links that have been associated with their role will be displayed on their home page. For example, an administrator may decide for a case worker security role, links to create a case, find a person, register a participant, change my password and search for a case, all should appear as quick links so that a caseworker always has access to them regardless of how they configure their home page.	• QUICKLINK • QUICKLINKAPPLICATIONLINK
User Interface Pods Configuration	The business object is used to create default views in Pod containers. Pod containers are typically used in 'Application Views' or home pages, where the user is presented with a summary of information that is relevant to their day to day job. The configuration describes the physical layout of the container and its content. For example, the configuration may describe a container with 3 columns and 6 available Pods, where 3 Pods are selected by default and displayed across the 3 columns.	• PAGECONFIG • USERPAGECONFIG

Application Modules

A list of the business objects that contain configuration data that relate to application modules and the entities that store the data. These include business object that relate to the following application modules: Verification, Evidence Broker, Appeals, Provider Management, Universal Access, and Outcome Management.

Note, appeals configuration is incorporated into the business object that the appeal configuration is associated with, a benefit product or an issue.

Table 12: Application modules business objects and entities

Business object	Description	Entities
Verification	Contains the configuration data for verification. Verification is the process of checking the accuracy of the information given by clients seeking services from a Social Enterprise organization. The verification of client information can take a number of forms; it can be provided by documents, for example, birth certificates or bank statements, or by verbal means, for example, telephone calls.	• VERIFICATIONCATEGORY • VERIFICATIONITEM • VERIFIABLEDATAITEM • VERIFICATIONITEMUTILIZATION • VERIFICATIONREQUIREMENT • DEPENDANTDATAITEM • VERIFICATIONREQUIREMENTUSAGE
Evidence Broker	Contains the configuration data for the evidence broker. The evidence broker provides a flexible approach to evidence sharing. Agencies can configure how they share evidence with other agencies in one way, and configure how they receive evidence broadcast from other agencies in a completely different way.	• EVIDENCEBROKERCONFIG • PRODUCTEVIDENCELINK • ADMINICEVIDENCELINK
Legal Action (Appeals)	Contains the configuration data for a legal action. A legal action is a type of case that defines legal processes associated with a case participant, involved with the organization that may include judicial involvement.	• ADMINLEGALACTION • ADMINLEGALACTIONLINK • ADMINLEGALSTATUS • CASEMULTILEGALSTATUSCONFIG • LEGALSTATUSCASETYPELINK • ADMINLEGALACTIONCATEGORY • LEGALCASETYPELINK

Provider Management

A list of the business objects that contain configuration data that relate to Cúram Provider Management and the entities that store the data. These include the following business objects: Attendance Payment Frequency, Roster Generation Frequency, Training, Certification, Service Group, and Service Offering.

Table 13: Application modules business objects and entities

Business object	Description	Entities
Attendance Payment Frequency	Contains the configuration data for the attendance-based payment frequency. This configuration allows an agency to set a default payment frequency for all services that are paid based on information included on attendance rosters.	• ATTENDANCEPAYMENTFREQUENCY
Roster Generation Frequency	Contains the configuration data for the attendance roster generation frequency. This configuration allows an agency to set a default roster generation frequency for all services for which attendance tracking is enabled..	• ROSTERGENERATIONFREQUENCY
Training	Contains the configuration data for a training program. A training program can represent a course completed by a provider, which may be required in order to deliver services.	• TRAINING • TRAININGCREDIT

Business object	Description	Entities
Certification	Contains the configuration data for a certification. A certification can represent a qualification completed by a provider, which may be required in order to deliver services.	• CERTIFICATIONCHANGEHISTORY • CERTIFICATION • ORGANISATIONUNIT
Performance Measure	Contains the configuration data for a performance measure. A performance measure defines the criteria by which the performance of a provider can be measured.	• PERFORMANCEMEASURE
Placement Payment Frequency	Contains the configuration data for the placement-based payment frequency. This configuration allows an agency to set a default payment frequency for all services that are paid based on placement information.	• PLACEMENTPAYMENTFREQUENCY
Service Evaluation Criterion	Contains the configuration data for a service evaluation criterion. This service evaluation criterion allows an agency to define the criteria by which the effectiveness of a service delivery to a client can be evaluated.	• SERVICEEVALUATIONCRITERION • RESPONSESCORE
Absence Reason Configuration	Contains the configuration data for an absence reason. This configuration allows an agency to define whether a client absent for a particular reason should have the units of service deducted from their units authorized, and whether or not the provider should still be paid.	• ABSENCEREASONCONFIGURATION
Service Group	Contains the configuration data for a service group. A service group is used to group a number of related services.	• SERVICEGROUP • SERVICEOFFERINGGROUPLINK

Service Offering

The Service Offering business object contains the configuration data for a service offering.
A service offering represents a service which may be offered by an agency to its clients. The configuration of the service offering defines key information about the service, for example the cost per unit.

The following entities store the configuration data that is contained within the business object.

• SERVICEOFFERING
• SERVICERATE
• PAYMENTOPTIONS
• SERVICEDELIVERYCONFIGURATION
• CREOLERULESETLINK
• SOSECURITYRIGHTS
• ABSENCEREASONCONFIGURATION
• SOATTENDANCECONFIGURATION
• SOATTENDANCEPAYMENT
• SOROSTERCONFIGURATION
• SOAPPROVALCRITERION
• CONTRACTTEXTLINK
• CONTRACTTEXT
• SOEVALUATIONCRITERION

- SOPERFORMANCEMEASURELINK
- SOTRAININGREQUIREMENT
- TRAININGSERVICEOFFERING
- SERVICEOFFERINGGROUPLINK
- LOCALIZABLETEXT
- TEXTTRANSLATION

Universal Access

A list of the business objects that contain configuration data that relate to Universal Access and the entities that store the data. These include the following business objects: Intake Application, Application Case, Screening, Triage, Life Event, and Motivation.

Intake Application

The Intake Application business object contains the configuration data for an intake application. An intake application allows a citizen to apply for programs by submitting an application that contains the required information on personal details, income, expenses, employment, education, and so on.

This information becomes evidence on the client's case and is used to determine eligibility.

The following entities store the configuration data that is contained within the business object.

- INTAKEAPPLICATIONTYPE
- APPLICATIONCATEGORY
- APPLICATIONCATEGORYLINK
- PDFFORM
- WITHDRAWALREQUESTREASON
- APPLICATIONTYPEPROGRAMLINK
- PROGRAMTYPE
- EXTERNALSYSTEM
- PDFFORMPROGRAMLINK
- SERVICEAREA
- SERVICEAREALINK
- PDFFORMOFFICESERVICEAREALINK
- MAPPINGCONFIGURATION
- PROGRAMOFFICESERVICEAREALINK
- ONLINEPROGRAMGROUP
- ONLINEPROGRAMGROUPLINK
- LOCALIZABLETEXT
- TEXTTRANSLATION

Application Case

The Application Case business object contains the configuration data for an application case. An application case is used to manage the information relating to a citizen's application for programs. This information is captured as part of an intake application. An intake application allows a citizen to apply for programs by submitting an application containing required information on personal details, income, expenses, employment, education, and so on.

When an intake application is submitted, an application case is created. The application case stores all the information submitted with the intake application.

The following entities store the configuration data that is contained within the business object.

- APPLICATIONCASEADMIN
- LOCALIZABLETEXT
- TEXTTRANSLATION
- CASECONFIGURATIONEVIDENCELINK

The application case business object is configured to use Cúram Express Rules which determine program eligibility and authorization processing. Additional configuration data includes the owner of the application and organization workflows used during application case creation and processing.

> **Note:** The application case business object has a dependency on Cúram Express Rules, organization, and workflow business objects. In order to successfully transport the application case business object change set, the organization, Cúram Express Rules, and workflow business objects must already exist on the target system before the application case business object is transported.

Screening

The Screening business object contains the configuration data for a screening. Screening allows a citizen to determine if they are potentially eligible for one or more programs based on a short set of guided questions and a short set of eligibility rules. Based on this determination, the client can decide whether to apply for the programs identified.

The following entities store the configuration data that is contained within the business object.

- SCREENINGTYPE
- SCREENINGRULESLINK
- SCREENINGCATEGORYLINK
- APPLICATIONCATEGORY
- LOCALIZABLETEXT
- TEXTTRANSLATION
- ONLINEPROGRAMITEM
- ONLINEPROGRAMGROUP
- ONLINEPROGRAMGROUPLINK
- PROGRAMTYPE
- MAPPINGCONFIGURATION
- PDFFORMPROGRAMLINK
- PDFFORM
- PROGRAMOFFICESERVICEAREALINK
- SERVICEAREA
- PDFFORMOFFICESERVICEAREALINK
- SERVICEAREALINK
- EXTERNALSYSTEM
- REMOTESYSTEM
- TARGETSYSTEM

Triage

The Triage business object contains the configuration data for triage. A triage process which can be used to quickly identify a citizen's needs and can direct them to community services and

government programs. For example, a client may have an immediate need for food and shelter for his or her family.

Triage helps to quickly identify these needs and provide details of suitable services, service providers and programs to help meet those needs.

The following entities store the configuration data that is contained within the business object.

- TRIAGECONFIGURATION

Life Event

The Life Event business object contains the configuration data for a life event. A life event is a situation that occurs in a citizens life which may result in a series of interactions between the citizen and the agency, for example, marriage, birth of a child, death.

Universal Access (UA) life events provides an agency with the capability of configuring life events which enable a citizen to report significant life changes which may impact the programs and services that the citizen is receiving or is due to receive.

The following entities store the configuration data that is contained within the business object.

- LIFEEVENTTYPE
- LIFEEVENTCONTEXT
- LIFEEVENTCATEGORY
- LIFEEVENTCATEGORYLINK
- LOCALIZABLETEXT
- TEXTTRANSLATION
- MOTIVATIONS
- RESULTSCATEGORYENTITY
- RESULTSPAGEACTION
- RESULTSCATEGORY
- LIFEEVENTCTXEVDMAPPINGCFGLNK
- LIFEEVENTCTXRMTLINK
- EVIDENCEMAPPINGCONFIGURATION
- REMOTESYSTEM
- TARGETSYSTEM
- EXTERNALSYSTEM

Outreach

The Outreach business object contains the configuration data for outreach campaigns. Outreach campaigns allow agencies to define targeted campaigns which can be displayed in the citizen account. For example, if the citizen is unemployed and has a child of school going age, information on childcare can be displayed, which will assist the citizen when they find work.

The following entities store the configuration data that is contained within the business object.

- CITIZENCAMPAIGN
- ADVICECONTEXTRULES
- ADVICECONTEXTCONFIGURATION
- IMAGE
- LOCALIZABLETEXT
- TEXTTRANSLATION

Motivation

The Motivation business object contains the configuration data for a motivation. A motivation is used to handle custom processes that are used to meet specific needs of citizens. For example, a Social Enterprise organization may wish to define a custom process that allows citizens to find appropriate health insurance options.

A motivation consists of a set of questions that are displayed to the citizen when the motivation is initiated, rules that are run on the data provided by the citizen, and a set of results which are displayed after the motivation is run.

The following entities store the configuration data that is contained within the business object.

- MOTIVATIONS
- RESULTSCATEGORY
- CATEGORYACTION
- LOCALIZABLETEXT
- TEXTTRANSLATION
- RESULTSPAGEACTION

Merative ™ SPM Social Enterprise Collaboration

A list of the business objects that contain configuration data that relate to Cúram Social Enterprise Collaboration and the entities that store the data. The business objects are the Social Enterprise Folder and Multidisciplinary Team Member.

Table 14: User interface business objects and entities

Business object	Description	Entities
Social Enterprise Folder	Contains the configuration data for a social enterprise folder (SEF). SEF is a type of case used to present a view of client and case information from multiple agencies and systems. Agency workers and multidisciplinary team members can use a SEF to securely share information and work together in order to achieve positive outcomes for clients and their families. Rather than having to access many systems across various agencies to gain a complete understanding of a family's case history, the SEF displays this information in one central location.	• SOCIALENTERPRISEFOLDERADMIN • LOCALIZABLETEXT • TEXTTRANSLATION
Multidisciplinary Team Member	Contains the configuration data for a multidisciplinary team (MDT) member role. A MDT role is played by an MDT member, for example, doctor, psychologist, agency worker. MDT members can be assigned to an outcome plan to leverage the experience of various disciplines to effectively manage the needs of clients	• MDTMEMBERROLE • MDTSHARINGCONFIGRESTRICTION • MDTSHARINGCONFIGURATION • LOCALIZABLETEXT • TEXTTRANSLATION

Outcome Management

A list of the business objects that contain configuration data that relate to Outcome Management and the entities that store the data. These include the following business objects: Outcome Plan,

Outcome Plan Action, Outcome Plan Goal, Outcome Plan Objective, Condition Data, and CER Assessment Configuration.

Outcome Plan

The business object contains the configuration data for an outcome plan. An outcome plan enables a user to run assessments to identify client needs and to plan activities such as services, actions or referrals, to address the needs identified by the assessments. An outcome plan is used to achieve particular goals and objectives which aim to assist clients in becoming independent of the social enterprise agency. The configuration data that is contained within this business object is stored on the following entities:

- EXPECTEDOUTCOMEPLANLINK
- OUTCOMEPLANADMIN
- OUTCOMEPLANADMINCASELINK
- OUTCOMEPLANGOALADMINLINK
- RECOMMENDATIONADMIN
- RECOMMENDATIONADMINCASELINK
- RECOMMENDATIONCONDITIONCASELINK
- CATEGORYCLASSIFICATION
- CLASSIFICATIONGUIDANCE
- FACTORLINK
- TEXTTRANSLATION
- LOCALIZABLETEXT
- SOCIALENTERPRISEFOLDERADMIN
- CASECOLLABORATIONADMINLINK
- VISITATIONCOMPLIANCYADMIN
- VISITATIONRESCHEDULEADMIN
- PRINTABLEDOCUMENTCONTENTADMIN

Outcome Plan Action

The Outcome Plan Action business object contains the configuration data for an outcome plan action. An action is a type of activity that can be added to an outcome plan to address needs identified by an assessment.

The following entities store the configuration data that is contained within the business object.

- OUTCOMEPLANACTIONADMIN
- LOCALIZABLETEXT
- TEXTTRANSLATION

Outcome Plan Goal

The Outcome Plan Goal business object contains the configuration data for an outcome plan goal. An outcome plan is used to achieve particular goals which aim to assist clients in becoming independent of the social enterprise agency.

The following entities store the configuration data that is contained within the business object.

- OUTCOMEPLANGOALADMIN
- LOCALIZABLETEXT
- TEXTTRANSLATION

Outcome Plan Objective

The Outcome Plan Objective business object contains the configuration data for an outcome plan objective. An outcome plan is used to achieve particular objectives which aim to assist clients in becoming independent of the social enterprise agency.

The following entities store the configuration data that is contained within the business object.

- EXPECTEDOUTCOME
- TEXTTRANSLATION
- LOCALIZABLETEXT

Condition Data

The Condition Data business object contains the configuration data for conditions. Conditions can be used to filter data such that it does not appear to a user, for example, recommended activities displayed in an outcome plan can be filtered such that only particular recommendations are displayed if configured conditions are met.

A condition can be based on gender, age, a rule set, a factor score or a factor classification.

The following entities store the configuration data that is contained within the business object.

- CONDITIONDATA

Condition Set

The Condition Set business object contains the configuration data for condition sets. Condition sets allow conditions to be grouped together. A condition can be based on gender, age, a rule set, a factor score or a factor classification.

Conditions can be grouped together into sets, for example, a condition set can contain a condition for gender and a condition for age.

The following entities store the configuration data that is contained within the business object.

- CONDITIONSET
- CONDITIONSETLINK

Outcome Plan Factor

The business object contains the configuration data for an outcome plan factor. An outcome plan factor is a domain or dimension which can be used to assess a client, for example, General Education Diploma.

An outcome plan factor allows a user to assess a client for a factor which does not form part of an assessment which is used to assess a client across a number of factors, that is, an outcome plan factor can be added to an outcome plan in a standalone manner.

The following entities store the configuration data that is contained within the business object.

- FACTOR
- FACTORCATEGORY
- FACTORCLASSIFICATION
- RECOMMENDATIONADMIN
- TEXTTRANSLATION
- LOCALIZABLETEXT

Outcome Management Questionnaire

The Outcome Management Questionnaire business object contains the configuration data for the questionnaire used by an assessment definition which uses a CDA matrix to produce assessment results.

The following entities store the configuration data that is contained within the business object.

- QUESTION
- QUESTIONNAIRE
- QUESTIONNAIREVERSION

Assessment

The Assessment business object contains the configuration data for an assessment definition. An assessment is used to identify and determine a client's needs, for example a Family Strength and Needs Assessment. An assessment assesses a client across a number of factors, for example, Substance Abuse, Income and Housing.

Based on the results of the assessment, activities can be recommended to address the needs identified by the assessment.

The following entities store the configuration data that is contained within the business object.

- ASSESSMENTDEFINITION
- ASSESSMENTDEFINITIONCASELINK
- ASSESSMENTROLE
- APPROVALCHECK
- CATEGORYCLASSIFICATION
- CLASSIFICATIONGUIDANCE
- FACTOR
- FACTORCATEGORY
- FACTORCLASSIFICATION
- FACTORLINK
- DAOUTCOMEFACTORLINK
- RECOMMENDATIONADMIN
- RECOMMENDATIONADMINCASELINK
- LOCALIZABLETEXT
- TEXTTRANSLATION

CDA Assessment Configuration

The CDA (Cúram Decision Assist) Assessment Configuration business object contains the configuration data for an assessment definition which uses a CDA matrix to produce assessment results. An IEG script is used to collect the answers to the assessment questions and a CDA matrix is used to produce results based on the answers to these questions.

The following entities store the configuration data that is contained within the business object.
'

- DAASSESSMENTCONFIGURATION
- DAOUTCOMEFACTORLINK
- DAOUTCOME
- DECISIONMATRIX
- DECISIONMATRIXVERSION

- DMOUTCOMELINK
- DMQUESTIONLINK

CER Assessment Configuration

The Cúram Express Rules (CER) business object contains the configuration data for an assessment definition which uses a CER rule set to produce assessment results. An IEG script is used to collect the answers to the assessment questions and a CER rule set is used to produce results based on the answers to these questions.

The following entities store the configuration data that is contained within the business object.

- CERTSPECIFICATION
- LOCALIZABLETEXT
- TEXTTRANSLATION

Federal Allowable Component

The Federal Allowable Component business object contains the configuration data for a federal allowable component. A federal allowable component is a component for which associated activities count towards a state's federal participation rate. Federal allowable components can be specified when activities are added to an outcome plan.

The following entities store the configuration data that is contained within the business object.

- FEDERALALLOWABLECOMPONENT
- FEDERALALLOWABLECOMPONENTLINK
- LOCALIZABLETEXT
- TEXTTRANSLATION

Cúram Family Services

Use this information to learn about the Cúram Family Services Suite and to understand what is contained in the following business objects: Contact Log, Placement Type Mapping, and Master Detail Code Table.

Table 15: Cúram Family services business objects and entities

Business object	Description	Entities
Contact Log	Contains the configuration data for a contact log. Contact log is used by caseworkers to document the actions they have taken, the individuals they have spoken to, what they have learned, conclusions they reached based on this information, and the basis or rationale for those conclusions. The contact log allows a caseworker the ability to capture details of events, interviews and/or meetings in a timely fashion so that the descriptions of those events are not influenced by later events.	• CONTACTCOMPLIANCECASETYPE • CONTACTCOMPLIANCEFACTOR • CONTACTCOMPLIANCEFREQUENCY
Placement Type Mapping	Contains the configuration data for a placement type mapping. Placement type mapping is a mapping between placement type and service offering that helps to simplify the process for completing placements by eliminating the need to identify both placement type and service offering before conducting a provider search.	• PLACEMENTSERVICEOFFERINGCONFIG

Business object	Description	Entities
Master Detail Code Table	Contains the data for a properties application resource. It can be used by the system in a variety of ways; some alter the functionality provided by the system and therefore they configure the system to suit an organization's needs.	• CODETABLECOMBOTYPE • CODETABLECOMBOITEM

Merative SPM Income Support Suite

Use this information to learn about the Merative SPM Income Support Suite and to understand what is contained in the following business objects: Summary Rules and Benefit Delivery.

Table 16: Merative SPM Income Support Suite business objects and entities

Business object	Description	Entities
Summary Rules	Contains the configuration data for the summary views. This data contains details of the rules class and attributes whose value need to be rendered. It holds the group definition and field details. The summary views are used only for classic rules.	• SUMMARYVIEW DEFINITION • SUMMARYVIEW DEFINITION • SUMMARYVIEW DEFINITION
Benefit Delivery	Contains the configuration data for the Children Health Insurance Program (CHIP) product delivery case type. It holds the county details, fee for service, primary care provider, and third-party product details for the CHIP case type.	• COUNTRYDETAILS • COUNTRYDETAILSLINK • FEEFORSERVICE • FEEFORSERVICELINK • PRIMARYCAREPROVIDER • THIRDPARTYPRODUCT • THIRDPARTYPRODUCT • LINK

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.