IBM Cúram Social Program Management
Version 7.0.5

*Cúram Configuration Transport Manager Guide*

IBM

**Note**

Before using this information and the product it supports, read the information in "Notices" on page 46

# Contents

# List of Figures

# Chapter 1. Cúram Configuration Transport Manager Guide

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.

## Main Features of the Configuration Transport Manager

An overview of the Configuration Transport Manager (CTM) which describes the business need for CTM and an overview of the CTM main features.

### The Business Need for CTM

A typical system landscape consists of a development system, test system and production system.

The Cúram Business Application Suite includes a wide breadth of administration and configuration data. Customers are likely to have the need to configure and test this on development and test systems ahead of deploying to production. CTM now provides the functionality to transport this 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. CTM provides a mechanism for packaging configuration data and transporting it from one system to another.

The following sample scenario illustrates how CTM works: A user creates or changes some configuration data on a development system. The user then uses CTM to transport the configuration data to a test system, which has a configuration that closely mirrors the production system. Once the configuration data has been successfully tested on the test system, the user uses CTM to transport the configuration data to the production system and make it live.

### Change Sets

A change set is the mechanism for packaging configuration data and transporting it from one system to another. A change set goes through multiple stages in its lifecycle. A user first creates a change set and then adds the configuration data that needs to be transported to the change set. The user releases the change set once the contents of the change set have been finalized. A released change set means that the contents of the change set cannot be modified. Once the change set is released, the user then transports the change set and configuration data to another system. The change set can then be applied in the system which means that the configuration data contained in the change set has been written to the database and therefore is live on that system.

### Business Objects

For the purposes of CTM we have defined a Business Object as a set of administrative data that defines and governs an instance of a particular set of functionality. Each Business Object consists of the collection of data (i.e. entity instances) that is required to configure the system to use and/or act on the instance of the functionality that it represents. It is through the capture, transport, and application of sets of Business Objects that CTM provides its capabilities.

All the business objects containing configuration data suitable for transporting, have been analyzed and integrated with CTM. A user can select an instance of one of these business objects and add it to a change set. The user does not require any further knowledge of the underlying data structure for the business object. A list of the Business Objects is provided in the appendix.

## Bi-directional Transport

CTM allows bi-directional transport of configuration data between systems in the system landscape. This facilitates flexibility whereby customers can transport configuration data between systems based on their business needs. For example, even when a customer ordinarily uses CTM for transporting their configuration changes from development through testing into production; they may have a need to apply a change outside of 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.

## Reverting a Configuration

Making configuration changes on a production environment is often considered a high risk activity. CTM provides undo functionality to mitigate against this risk. Undo functionality allows a customer to revert the system back to the configuration prior to applying the 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 prior to applying the change set. After a user applies a changeset, they can select the undo action for the change set, which will revert the system back to the original configuration by applying the contents of revert change set.

It must be remembered however, that the undo option is only useful for a limited period of time. Once the application starts to use the configuration data, runtime data is created based on the latest set of configuration data. At this stage, to revert to an earlier configuration requires reverting the runtime data also and in this case a database rollback is necessary as otherwise the configuration data and runtime data would be out of sync.

## Audit Controls in CTM

CTM provides a number of features in support of accountability and traceability. The name of the change set must be unique in support of tracking the change set. The contents of a change set are frozen before it can be moved across systems and therefore the contents of an applied change set will always be known. Other history information such as when it was transported or applied is also maintained to provide an audit trail.

# Change Set Lifecycle

Change sets provide the means to package and transport groups of configuration changes in the form of business objects from one system to another. Change sets can also be used for reverting to a defined set of business objects at a later point in time.

## Change Set Definition

### Creating a Change Set

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

### Adding Business Objects to a Change Set

The content of a change set can be defined by a user or a group of users when the change set is in an 'Open' state. That is, a user or a group of users identify the business objects that need to be transported to a target system and adds them to the change set. A search feature is provided whereby users can search for business objects and add them to a change set. Business objects can be added or removed from the change set when the change set is in an 'Open' state.

### Adding Related Business Objects to a Change Set

The "Add Related Business Objects" feature provides guidance to a user for business objects that might also need to be added 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.

This feature assists customers in transporting the correct changes to ensure data integrity on the target systems. The business object along with its related business objects is required to maintain the business object configuration data and relationships in the target system exactly as it exists 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 lots of 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 via database relationships; i.e. foreign key relationships or link tables. If any other type of relationship exists between business objects, this will not be reflected in 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 but this won't be presented to the user as a dependent business object. Therefore 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 a user 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. Once released, the change set can be transported to a target system. The released change set can also be used to revert to the configuration data associated with the business objects in the change set at a later point in time.

## Transporting Business Objects

### Transporting a Change Set

A released change set can be automatically transported by selecting a target system to which the change set will be transported. This utilizes the web services feature provided by the application. For more details, refer to the Target System chapter in System Configuration guide for further details on how to setup or maintain the web services feature.

Released change sets can also be manually transported to a target system. Manual transport is used when it is not possible to create a connection between two systems over a network. For example, there may be network issues or security policies may mandate that the production systems need to be hosted on networks 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.

**Note:** With a manual transport, there is an option to encrypt the change set. The method used is Base64 which provides an obfuscation mechanism rather than encryption.

### Applying a Change Set

User can apply a released change set, whereby the configuration data in the business objects in the change set is made live in the database. Applying the change set can result in the addition of business objects in the target system, if the business objects did not already exist on the target system or the modification of business objects in the target system if the business objects already existed. The change set can then be transported to another system once the applied changes are successfully tested on the target 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.

A change set can also be created and applied in the same system for the purpose of reverting to an earlier working configuration data if application issues are discovered. Refer to "Reverting to a Defined Set of Configuration Data" on page 5 for more details.

## Other Change Set features

### Cloning a Change Set

Change sets can be cloned but the new version of the change set must have a new name for traceability purposes. The content can then be modified to meet changing requirements. The cloned change set will have a status of Open and will contain the list of business objects that existed in the change set that was cloned. Business objects can then be added or removed from the change set as required. This helps to create a change set with business objects similar to an existing change set automatically rather than creating a change set manually and then adding the required business objects to the change set.

When a change set is cloned, only the identifiers of the business objects are cloned. The data contained in the business objects is not cloned. When the new change set that has been created through the clone is released, it will contain any changes that have been made to the business objects up to the point when the new change set is released. Therefore, if any changes have been 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 differ.

### 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 will change the status of the change set to Cancelled.

## Dealing with Configuration Transport Errors

As using the CTM often involves modifying configuration data in production systems, 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 of CTM

CTM can be used to transport configuration data changes between systems in a structured manner leading to the modification of configuration data in a production system. It supports the testing of configuration changes on development and test environments prior to deploying them on a production system. It does not however 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 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 CTM can be used incorrectly, it is recommended that customers incorporate the use of CTM 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.

However even with the tightest release process, things may still occasionally go wrong and therefore the remainder of this chapter describes features that can be useful in such scenarios. Please refer to *section 5.2.2 - Data Management Policy* for more information on recommended procedures for managing Business Object Creation and Modification.

## Reverting to a Defined Set of Configuration Data

As discussed in Chapter 2, CTM includes a feature which supports reverting back to the configuration that existed on the system prior to applying the new change set. A revert change set is created when applying a change set on a system. For each business object listed in the change set, a snapshot of that business object as it currently exists on the system prior to be change set being applied is taken. This snapshot is added into a revert change set. The revert change set is automatically created in a released state, meaning the change set is in a state that it can be applied. The contents are preserved should the system need to revert back to this configuration. This revert change set is not presented in the user interface but it is applied on the system when the user selects 'undo' action on the original change set.

If issues are encountered after applying the transported change set then the revert change set can be applied on the target system by selecting the undo action, to revert to the earlier set of configuration data. After reverting, the system is returned to the state it was in before applying the transported change set. 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 will be reverted back to the previous state using the business objects in the automatically created change set.

**Note:** When logically deletable business objects are reverted, they are logically deleted and not physically deleted. Therefore, 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 beforehand.

As mentioned previously, there is only a certain period of time during which the option to revert configuration data is feasible. Once live runtime data is created based on the newly transported configuration data, the revert feature can no longer be used as otherwise the configuration data and runtime data are out of sync.

## Accountability

CTM provides accountability with respect to the various operations performed to transport business objects from one system to another. The following sections describe these features.

### Change Set Transport History

Every 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 3 system landscape for development, testing, and production, the following entries are created when transport "Release 0618" change set. When the change set is transported from development to testing, 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 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.

**Change Set Undo History**

When a change set is applied, if it is subsequently reverted by selecting to undo the change set, an entry is logged in the undo history for that change set. The undo history includes the user who initiated the undo action and the date and time on which it occurred.

With the transport history, apply history and undo history available, a user can gain a full picture of which systems a change set has been transported to, if it was applied on these systems and if it were subsequently rolled back on any of these systems.

# Setting Up Configuration Transport Manager

To set up systems to use CTM, you must complete the prerequisite activities, the procedure to set up CTM on new application installations, and the procedure to enable the use of CTM on existing application installations. You can also complete some other administrative activities on the systems where CTM is in use.

## Prerequisites

There are some prerequisite activities that must be carried out before commencing setup and use of CTM. These activities are described in this section.

**Define the System Landscape**

*Overview*

The system landscape consists of the set of the systems among which data can be transported using CTM. One of the primary tasks that must be performed in advance of setting up and using CTM is to define the initial structure of the system landscape. That is, to decide on which systems the system landscape will contain, to decide on the role of each system (e.g. Development, Test, Production), and to decide on the transport paths between the systems.

*Recommended Minimum Configuration*

It is strongly recommended that at a minimum, a system landscape should consist of three systems, one for each of the following roles: Development, Test, and Production.

The precise activities carried out on each of the systems in the above minimum system landscape configuration are as follows:

- *Development System:* The development system is the system on which business objects are initially created and subsequently changed. When the administrative user is satisfied that the business objects are configured correctly, they are transferred and applied to the test system using CTM.
- *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. Note that to avoid consistency issues, business objects should not generally be created or changed on the test system - please refer to *section 5.2.2 - Data Management Policy* more information on policies for creating and changing business objects. When it has been verified that the business objects are behaving as expected, they can be transferred and applied to the production system.

- *Production System:* The production system is the live system 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 needs to be taken before deciding to create or change a business object on the production system. Please refer to *section 5.2.2 - Data Management Policy* for more information on policies for authoring and changing business objects.

### Other System Landscape Configurations

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 may be required. Another possibility is to have a system landscape consisting of a development system, a test system, a training System, and a production system.

However, 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's also recommended that the number of systems in a system landscape is not allowed to grow too large. This is due to the difficulties that may arise in managing the transport paths between the systems, and the associated problems that may arise with regard to system consistency.

### 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 may cause problems when applying and/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(s).
- *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.

### Adding Systems to a System Landscape

Further systems can be added to a system landscape after it has been initially setup. Please refer to *section 5.5.1 - Adding Systems to an existing System Landscape* for more information.

### Landscape Name

A system landscape needs to have a unique name. Each system in the system landscape will be configured with this name. So, some thought needs to be put into selecting an appropriate name for the system landscape. As an example, the system landscape name could be the organization name.

There are some restrictions on the 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.

- The landscape should not be named *nolandscape*. This name has special significance - see *section 5.3.1 - Landscape Name* for more details on the *nolandscape* landscape name.

### Multiple System Landscapes

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.

### Data Management Policy

A policy on how to manage administrative data within the system landscape must be defined. Some of the key elements of this policy are described in this section.

### Business Object Creation and Modification policy

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

*Option 1: Use Standard Development Lifecycle Practices for all Business Object Types*

The preferred option is that all business objects should be managed using standard development lifecycle practices. That is, business objects should be created or modified only on the development system(s) in a system landscape. So, business objects should not be created or modified on test or production systems. Instead, the new or modified business objects should be transported from the development system to the test system and then, following testing, transported to the production system. In addition to helping ensure that there are no issues with a particular business object, this policy removes the potential for transported business objects to overwrite local changes on production systems, and reduces the possibility of clashes that may prevent a change set from being applied.

*Option 2: Manage some Business Object Types locally*

In some cases, the policy described above will not be sufficiently flexible. For example, it may 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 considered too high.

In this case, the recommendation is that a clear distinction is made between business object types are going to be managed locally, and those that will be managed using standard development lifecycle practices, leveraging CTM as detailed above. CTM should then only be used with business objects where the business object type is being managed using standard development lifecycle practices. It should not be used to transport business objects where the business object type is being managed locally on the production system.

### Applying Change Sets to a Production System

Another element of the data management policy is to decide when change sets can be applied to the production system. It is recommended that change sets are only applied 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. This in turn reduces the potential for negative impact from unexpected issues arising due to the changes.

### 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.

There are two procedures that may lead to divergence between the test system business objects and the production system business objects:

1. If change sets containing business objects are transported and applied on the test system, but never transported and applied to the production system, then 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. Please note that this practice of adding or changing business objects directly on the production system is strongly discouraged - see the section *Business Object Creation and Modification policy*, above, for further details.

In order to avoid the possibility of substantial differences between the test system and production system emerging, it is recommended that a policy is adopted 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 has passed 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 will help in ensuring that the test system does not contain invalid configurations that can never be migrated to the production system.

### Key Block Range Allocation management

A unique key block range must be assigned to each system in the system landscape. This is in order to avoid the possibility of clashes in primary keys when a business object is transported and applied. The process is described in detail in *section 5.3.2 - Key Block Ranges*, below.

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

### Data Relating to Cryptographic Configurations

Because Cúram cryptographic configurations can be upgraded (see *Cúram Security Handbook*) you need to be aware of the configuration data in this regard, which may differ between source and target systems.

For out-of-the-box Cúram this data will be 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 cipher cryptographic setting differences across systems.
- Properties in the Properties table (via `Application.prx`) that contain password values - subject to cipher cryptographic setting differences across systems.

For the above you will have to re-digest and/or re-cipher the password data if there are source and target differences in the cryptographic configurations. See the *Cúram Server Developer's Guide* for more information regarding the Apache Ant targets for doing this.

## Setup Tasks

This section describes the tasks involved in setting up a new system with CTM enabled.

### Landscape Name

Each system in the system landscape needs to be configured with the landscape name. Change sets can then be transported to and applied on other systems configured with the same landscape name. The

landscape name configuration is performed using the CTM setup tool. See *section 5.3.4 - CTM Setup Tool* for details on how to use this tool.

### Default Landscape Name

If no landscape name is explicitly configured for a system, a default landscape name of *nolandscape* is used. There is a restriction on the use of CTM on systems with the landscape name *nolandscape*. The restriction is that change sets created on systems with the *nolandscape* landscape name cannot be applied on any other system, including other systems configured with the *nolandscape* landscape name. This means that in order to make full use of the capabilities offered by CTM, the system must be explicitly configured with a landscape name that differs from the default *nolandscape* landscape name.

### Key Block Ranges

Each system in the system landscape must be assigned a unique key block range. This range is used to ensure that when a business object is transported to another system in the system landscape, the primary keys and other unique ids used by the entities that make up the business object do not clash with those of entities already present on the target system. This subsection provides information on configuring systems with a unique key block range. In advance of reading this subsection, the chapter on *Unique IDs* in the *Cúram Server Developer Guide* should be read - it provides details of how unique IDs are used within the application.

### Key Block Range allocation

The key block range for a system is specified using two components: a group number and a range number. The group number is a number between 3 and 32,767 inclusive (group numbers 1 and 2 are reserved for existing data and for application usage, so customer configurations must start with group 3). The range number is a number between 1 and 512 inclusive. Each system in the system landscape must be configured with a unique group range pair. For example, the following configuration could be used for a three system landscape:

- production system: *group 3, range 2*
- test system: *group 3, range 3*
- development system: *group 3, range 4*

Alternatively, the following could be used:

- production system: *group 3, range 2*
- test system: *group 4, range 2*
- development system: *group 4, range 3*

It must be emphasized that it is very important that no systems in the system landscape are configured with the same group range pairs. Please refer to the *Key Block Range Allocation management* subsection of *section 5.2.2 - Data Management Policy* for details of a recommended policy for managing key block range allocations.

Key Block Range Scope

Key block range allocations are scoped within system landscapes. So, it is possible to have two different systems, each in a different system landscape, with the same key block range allocation. This is possible because it is not permitted to transport and apply business objects between systems in different system landscapes.

### Setting up the Key Block Range allocation for a system

It is important that key block ranges are explicitly configured on systems before they are first started. This is because after a system is started for the first time, the key block range allocated to it cannot be removed. This restriction is necessary because as soon as a system is started, entities drawn from the key block range may be created, and so removing a key block range allocation could cause issues.

The key block range configuration for a system is performed using the CTM setup tool. See *section 5.3.4 - CTM Setup Tool* for information on using this tool.

*Default Key Block Range*

There is one exception to the requirement to explicitly configure a key block range before a system is first started. This is that it is possible to have a single system in a system landscape that does not have a key block range explicitly configured before it is first started. This is because if no key block range is explicitly configured, a default key block range of *group 3, range 2* is used. Therefore, it is possible to have a single system in a landscape which is not explicitly configured with a key block range before it is first started, as it will use the default key block range allocation *group 3, range 2*.

However, every other system in the system landscape will need to have a key block range explicitly configured, and no other systems can be configured with the key block range *group 3, range 2*.

Note that the reason for supporting a default key block range allocation is to facilitate the enabling of CTM on a production system after a technical upgrade. Please refer to *section 5.4 - Enabling use of CTM on an existing application installation* for further information.

### Multiple Ranges

Each key block range (i.e. each *group,range* pair) contains a very large number of key blocks (over 4 billion). In normal usage, 256 keys are allocated from each key block, so the number of keys available from each range allocation is very large. Furthermore, each business object generally uses a separate key set (or collection of key sets), and so effectively has a separate "key counter" for use by its entities. This in turn means that keys are used up at an even slower rate than might otherwise be the case. Therefore, it is unlikely that key block range allocations will be exhausted in normal usage.

Nonetheless, to cater for the possibility of key block range exhaustion, a system can be configured with multiple key block ranges. Furthermore, these additional key block ranges can be added later if necessary - i.e. they can be added even after the system has first been started. If a system is configured with multiple key block ranges, it will transparently start using key blocks in the additional ranges when all key blocks in the original range have been used up.

Note that any additional key block range allocations must also be unique to the system - that is, they cannot be used on other systems within the system landscape. Also, as already noted, after a system has been configured with a particular key block range, it is not possible to remove that key block range from the system.

As an example, a production system could start with the allocation *group 3, range 2*. Then, later on, an additional allocation *group 3, range 44* could be added if required. If all Key Blocks in the allocation *group 3, range 2* are then used up, the system will start using keys from key blocks in the allocation *group 3, range 44*.

One point to note is that a log message will be output when a key set on the system is approaching the end of the range allocated to it. In the default system configuration, this message is output when the key set has 100,000,000 (one hundred million) key blocks left. This should allow for adequate time to provision the system with another key block range allocation in advance of the existing allocation being exhausted.

### CTM System Identifier

A unique human-readable name known as the CTM system identifier must be assigned to each system in the landscape. The CTM system identifier is used by the CTM user interface to indicate the source system on which a change set originated.

There are some restrictions on the value selected for the CTM System Identifier:

- The CTM system identifier should consist only of alphanumeric characters, with no spaces.
- The maximum length of the CTM system identifier is 500 characters, but it is recommended that a smaller identifier is used.
- The CTM system identifier should not be named *noname*. Please see the *Default CTM System Identifier* section below for further information.

Details on how to configure the CTM system identifier are provided in *section 5.3.5 - Configuring the CTM System Identifier*

### Default CTM System Identifier

If no CTM system identifier is explicitly configured for a system, a default value of *noname* is used. Systems using the default *noname* value are restricted in that change sets created on these systems cannot be released. This means that in order to make full use of the capabilities offered by CTM, the system must be explicitly configured with a CTM system identifier that differs from the default *noname* value.

### CTM Setup Tool

The CTM setup tool is used to configure both the landscape name and the key block range allocation for each system in the system landscape.

An Extensible Markup Language (XML) configuration file is used to specify the appropriate values. When executed, the CTM setup tool writes the configured values to the database so that they can be used by the system. As has previously been noted, it is important that the values are configured and the tool executed before the system is started. Please refer to *section 5.3.2 - Key Block Ranges* for further information on this requirement.

### Configuring Key Range Values and Landscape name

The configuration file is used to specify both the landscape name and the key block range allocation. The configurations are specified in XML.

The default configuration file location is *<CuramSDEJ>/scripts/config/rangeawareserver_config.xml*. The values in this file can be edited directly, or the file can be copied to an alternative location and edited. If an alternative location is used, then the location must be specified when executing the CTM Setup tool - please see below for instructions on how to do this.

The configuration file contains two XML stanzas: a stanza for specifying the landscape name, and a stanza for specifying the key block range values.

*Landscape Name Configuration Stanza*

The *<Landscape>* stanza of the configuration file is used to specify the landscape name. The default value for the landscape name is *nolandscape*. However, this should be changed to the name selected for the landscape. An example configuration is illustrated below:

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

*Figure 1: Sample Landscape Stanza*

*Key Block Range Configuration Stanza*

The *<Rangevalue>* stanza is used to configure the key block range allocated to the system. As noted in *section 5.3.2 - Key Block Ranges*, the key block range consists of two elements: a group number and a range number. Both are specified in the *<Rangevalue>* stanza, as illustrated in the example below:

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

*Figure 2: Sample Range Stanza*

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 these are required. Please refer to the *Multiple Ranges* sub-section of *section 5.3.2 - Key Block Ranges* for further information.

Note that both the group number and the range number are validated by the tool to ensure that the values are acceptable. Please refer to *section 5.3.2 - Key Block Ranges* for information on valid values.

*Executing the CTM Setup Tool*

The CTM setup tool is an Apache Ant target. The target is executed as part of the standard Ant *database* target, but can also be executed independently.

The normal procedure is to set the appropriate values for the landscape name and the key block range allocation in the configuration file before performing the database build. The configured values will then be loaded into the system as part of the database build. That is, they will be loaded by executing *build database*.

Alternatively, the configured values can be loaded into the database separately, by invoking the *insertrangeawareconfig* target - that is, by executing *build insertrangeawareconfig*.

If an alternative location to the default location is used for the configuration file, this should be specified by setting the system property *rangeAwareConfig* when invoking Ant. For example, the *database* target would be executed as follows:

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

*Figure 3: Specifying an alternative configuration file location when invoking the database target*

the *insertrangeawareconfig* target would be executed using:

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

*Figure 4: Specifying an alternative configuration file location when invoking the insertrangeawareconfig target*

**Configuring the CTM System Identifier**

The CTM system identifier is a unique human-readable name that must be configured for each system in a system landscape. Please refer to *section 5.3.3 - CTM System Identifier* for further information on the CTM system identifier requirements.

The CTM system identifier is configured using the property *curam.ctm.systemIdentifier*. This property is set using the system administrator *Property Administrator* functionality in the application. Note that the property is contained in the *Admin - System Settings* category. After the value has been set, it must be published in order for it to be available to the system.

It is highly recommended that once set, this property value is not changed again. This is because changing it may cause confusion with regard to the origin of transported change sets when they are viewed on a target system.

## Enabling use of CTM on an existing application installation

In some circumstances, a customer may wish to commence production use of the application without initially making use of CTM. This strategy avoids the need to initially consider CTM - specific concerns such as system landscape configuration, data management policies, key block range allocation, etc. Consideration of these issues can then be deferred until the decision is made to commence using CTM.

An example of when this strategy might be adopted is when an existing customer initially upgrades to a distribution of the application that supports CTM. In this case, the customer may 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 CTM.

**CTM Restrictions**

When a system is installed and started without performing the setup steps for CTM, it will have the default landscape name *nolandscape*, the default key block range allocation *group 3, range 2*, and the default CTM system identifier *noname*.

With these default settings, CTM can only be used in a restricted form. The restriction is that change sets created on the system cannot be transported and applied on any other system. So, a production system setup without performing the CTM setup steps will only allow for usage of CTM subject to these restrictions.

There is further information on the default values for these fields and the associated behaviour in *section 5.3 - Setup Tasks*.

**Setup Procedure Recommendations**

In order to allow for full use of all CTM features, a setup procedure must be implemented. This procedure is described in the subsections that follow. It is recommended that the setup procedure is first trialled on a non-live duplicate of the production environment before it is carried out on the production environment. Furthermore, it is suggested that all of the activities that involve making changes to the production system are carried out during scheduled maintenance time, when the production system is offline.

**Prerequisite Activities**

Before starting, a number of prerequisite activities need to be carried out. These are detailed blow, and are in addition to the pre-requisite activities detailed in *section 5.2 - Prerequisites*.

***Decide on Landscape Name***

The landscape name must be defined. The requirements for the landscape name are described in the *Landscape Name* subsection of *section 5.2.1 - Define the System Landscape*. Note that although the production system is initially configured with the default landscape name (*nolandscape*), it will be necessary to change this. The process for making this change is detailed below.

***Decide on Key Block Range allocations***

The production system will already have the default key block range allocation of *group 3, range 2*. However, every other system that is going to be in the same system landscape as the production system must also have a unique key block range allocation. So, these allocations need to be decided on and assigned. Note that in making the allocations to the other systems, the key block range group 3 range 2 should be avoided as this has already been assigned to the production system. Please refer to *section 5.3.2 - Key Block Ranges* for a description of the process involved in assigning key block ranges.

***Decide on CTM System Identifiers***

Each system must be provided with a unique CTM system identifier. The production system will initially be configured with the default CTM system identifier (*noname*), but this must be changed in order to fully enable CTM.

***Setup Procedure***

The setup procedure is divided into two sections - the procedure for setting up CTM on the existing production system, and the procedure for setting up CTM on the other systems in the system landscape.

**Production System Setup**

In order to enable usage of CTM on the existing production system, some configuration changes need to be made.

***Change Landscape Name***

The landscape name on the production system must be changed from *nolandscape* to the new name that has been selected. This will make the production system a member of the landscape, enabling full use of all CTM functionality without the restrictions outlined in *section 5.4.1 - CTM Restrictions*. Note that

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. This property can be changed using the system administrator *Property Administrator* functionality in the application. Note that the property is contained in the *Admin - System Settings* category. After the value has been changed, publish the changes so that they are made available to the system.

It is very important that the landscape name is not changed other then when performing this procedure to enable CTM 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.

### Setting the Production System CTM System Identifier

The CTM system identifier for the production system must be changed from the default value of *noname* to the new value that has been selected. Instructions for setting this are described in *section 5.3.5 - Configuring the CTM System Identifier*.

### Setting up other Systems in the System Landscape

The other systems in the system landscape also need to be setup - that is, the development system, test system, and any other systems required. These are required to be fresh installations of the application with CTM fully enabled at install time. The steps involved in the setup process are detailed below.

### Install 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. It's worth noting that this means that the installation should also include any customer customizations that have been developed and applied to the application. Please refer to *System Homogeneity* subsection of *section 5.2.1 - Define the System Landscape* for further information on the homogeneity requirements.

### Initial Data Load on 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 being the same as that on the production system. This is both so that changes can be made to pre-existing business objects on the other systems in the system landscape and so that any testing carried out on the development system or test system is valid.

To this end, the production system data for the entities that form the business objects needs to be copied over to the other systems in the system landscape before the other systems are first started. These entities are listed in Appendix A. It is recommended that this copy is made during scheduled maintenance time, when the production system is offline. This will help in ensuring 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 CTM that are used for all new application installations must be implemented. These are to setup the systems with the selected landscape name, set up the key block range allocations, etc. These steps are described in *section 5.3 - Setup Tasks*.

## Other Administrative Considerations

This section provides information on some other administrative activities that may be carried out on the systems where CTM 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 will need to 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 *section 5.4.5 - Setting up other Systems in the System Landscape.*

It is worth emphasizing that the key block range allocated to the new system must not previously have been assigned to an existing system in the system landscape.

# Physical Table to Admin Business Object Mapping

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

**Note:** The business objects listed here are available only in CTM if a suitable license is in place for the area in question.

## IBM Cúram Social Program Management Platform

Use this information about the configuration data that relates to the specified business objects to learn about the entities where the configuration data is stored for the various business objects.

**Case**

### *Benefit Product*

The 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 configuration data that is contained within this business object is stored on the following entities:

- 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 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 configuration data that is contained within this business object is stored on the following entities:

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

### Liability Product

The 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 configuration data that is contained within this business object is stored on the following entities:

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

- LOCALIZABLETEXT
- TEXTTRANSLATION

### Assessment

The 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 configuration data that is contained within this business object is stored on the following entities:

- ASSESSMENTCONFIGURATION
- ASSESSMENTRULESLINK

### CER Benefit Product

The 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 behaviours 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 (e.g. 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, we recommend that the following object types get published 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 Product Name code table must also be transported as a dependent business object.

The configuration data that is contained within this business object is stored on the following entities:

- 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 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 behaviours 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 Product Name and Product Type code tables must also be transported as a dependent business object.

The configuration data that is contained within this business object is stored on the following entities:

- 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 business object contains the configuration 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 configuration data that is contained within this business object is stored on the following entities:

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

### Investigation

The business object contains the configuration data for an investigation case type. Investigation cases are used to inquire into circumstances that surrounding 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 configuration data that is contained within this business object is stored on the following entities:

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

### Issues

The 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 configuration data that is contained within this business object is stored on the following entities:

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

### *Screening*

The 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 configuration data that is contained within this business object is stored on the following entities:

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

## Cúram Express Rules

### *CER Rule Set*

The 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 behaviours apply to a 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 (e.g. 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, we recommend that the following object types get published 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 configuration data that is contained within this business object is stored on the following entities:

- CREOLERULESETEDITACTION
- CREOLERULESETCATEGORYLINK

### CER Rule Object Propagator

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

**Note:** The following special behaviours 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 (e.g. 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, we recommend that the following object types get published 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 configuration data that is contained within this business object is stored on the following entities:

- RULEOBJPROPCONFIGSANDBOX
- LOCALIZABLETEXT
- TEXTTRANSLATION

### CER Rule Set Category

The business object contains the configuration data for the CER rule set category. The CER rule set category describes and categorizes CER Rule Sets. The configuration data that is contained within this business object is stored on the following entities:

- CREOLERULESETCATEGORY
- LOCALIZABLETEXT
- TEXTTRANSLATION

**Organisation**

### Organisation Structure

The business object contains the configuration data for an organization structure. The organization structure provides a means of managing the organization's users and functional areas. More specifically, the organization structure allows for the partition of administration functionality related to an organization's hierarchy structure across many lines. The configuration data that is contained within this business object is stored on the following entities:

- ORGANISATIONSTRUCTURE
- ORGANISATION
- ORGANISATIONUNIT
- ACTIVITY
- ACTIVITYRECURRENCE
- ORGUNITPARENTLINK
- ORGUNITPOSITIONLINK
- POSITION

- JOB
- ORGANISATIONUNITRESOURCE
- RESOURCES
- POSITIONREPORTINGLINK
- POSITIONLOCATIONLINK
- CASEAPPROVALCHECK
- TEMPORALEVIDENCEAPPROVALCHECK
- INVESTIGATIONAPPROVALCHECK
- ISSUERESOLUTIONAPPROVALCHECK
- MILESTONEWAIVERAPPROVALCHECK
- EVIDENCEAPPROVALCHECK
- SECURITYIDENTIFIER
- LOCALIZABLETEXT
- TEXTTRANSLATION

### *Location Structure*

The business object 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. The configuration data that is contained within this business object is stored on the following entities:

- LOCATIONSTRUCTURE
- LOCATION
- WORKINGPATTERN
- ADDRESS
- LOCATIONACCESS
- LOCATIONHOLIDAYLINK
- LOCATIONHOLIDAY
- LOCATIONRESOURCELINK
- RESOURCES
- DAILYSCHEDULE
- SLOT
- EMAILADDRESS
- POSITIONSLOTAVAILABILITY
- PHONENUMBER
- SECURITYIDENTIFIER

### *User*

The business object 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. The configuration data that is contained within this business object is stored on the following entities:

- USERS
- PHONENUMBER
- EMAILADDRESS
- USERSKILL
- RESOURCES
- ACTIVITY

- ACTIVITYRECURRENCE
- TASKREDIRECTION
- CASEAPPROVALCHECK
- EVIDENCEAPPROVALCHECK
- SERVICEPLANAPPROVALCHECK
- TEMPORALEVIDENCEAPPROVALCHECK
- INVESTIGATIONAPPROVALCHECK
- ISSUERESOLUTIONAPPROVALCHECK
- MILESTONEWAIVERAPPROVALCHECK
- WORKINGPATTERN
- POSITIONHOLDERLINK

### External User

The business object 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 those of internal users. The configuration data that is contained within this business object is stored on the following entities:

- EXTERNALUSER
- EXTERNALUSERPARTICIPANTLINK

## Dynamic Features

### Dynamic EvidenceType

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 configuration data that is contained within this business object is stored on the following entities:

- 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. The configuration data that is contained within this business object is stored on the following entities:

- EVIDENCETYPEVERSIONDEF
- EVIDENCERULESETDEF

### Dynamic Product Evidence Type Link

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

**Note:** The following special behaviour applies to a dynamic product evidence type link business object: Transported links between products and dynamic evidence types will not be seen on the target system

unless the dynamic evidence type has at least one dynamic evidence type version with a status of 'active'. However, when dynamic evidence type versions are transported, they arrive in the target system as 'InEdit'. As such, 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, and activate the transported dynamic evidence type versions; then the administrator can transport the relevant link business object.

The configuration data that is contained within this business object is stored on the following entities:

- PRODUCTEVIDENCETYPEDEFLINK
- PRODUCT
- EVIDENCETYPEDEF

### Integrated Case Evidence Type Link

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

**Note:** The following special behaviour applies to an integrated case evidence type link business object: Transported links between integrated cases and dynamic evidence types will not be seen on the target system unless the dynamic evidence type has at least one dynamic evidence type version with a status of 'active'. However, when dynamic evidence type versions are transported, they arrive in the target system as 'InEdit'. As such, 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, and activate the transported dynamic evidence type versions; then the administrator can transport the relevant link business object.

The configuration data that is contained within this business object is stored on the following entities:

- ADMINICEVIDENCETYPEDEFLINK
- ADMININTEGRATEDCASE
- EVIDENCETYPEDEF

### Case Audits

### Case Audit Configuration

The business object 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. In addition to compliance monitoring, a range of information on quality, practice, and outcomes can also be gathered and assessed.

**Note:** The following special behaviour applies to a case audit configuration business object: Case audit configurations cannot be removed as every case type should have a configuration.

The configuration data that is contained within this business object is stored on the following entities:

- AUDITCASECONFIG
- AUDITCASEFOCUSAREA
- AUDITCASESELECTIONQUERY

### Case Audit Selection Query

The business object contains the configuration data for selection queries. 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:** The following special behaviour applies to a case audit selection query business object: Active Selection Queries cannot be removed.

The configuration data that is contained within this business object is stored on the following entities:

- SELECTION QUERY
- CRITERIA
- SELECTIONQUERYCRITERIALINK

**Business Intelligence and Reporting Tools**

*Report Configuration*

The business object contains the configuration data for the Business Intelligence and Reporting Tools (BIRT) reports. This specifies the configuration for displaying a report. Multiple configurations can be created for each report, using different logical names, allowing 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:** The following special behaviour applies to a BI report business object: When transporting BI viewer configuration details, its root parameter is not transported as the target system most likely will have a different root. The system administrator must set the root parameter on the target system, after the transportation.

The configuration data that is contained within this business object is stored on the following entities:

- BIREPORTCONFIGURATION
- BIREPORTPARAMETER

*Viewer Configuration*

The business object contains the configuration data for the BI viewer details. This allows the user to change from the default viewer, which is the BIRT viewer, to use the Actuate BIRT viewer.

**Note:** The following special behaviour applies to a BI viewer business object: In order for transported BI report configuration to be functional the target system must have BI viewer configuration set-up.

The configuration data that is contained within this business object is stored on the following entities:

- BIVIEWERCONFIGURATION
- BIREPORTPARAMETER

**Workflow**

*Workqueue*

The business object 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 using the Workflow Administration functionality where an administrator can create, edit or remove work queues from the system. An administrator may subscribe users to the work queue or the users themselves can subscribe to the work queue. The configuration data that is contained within this business object is stored on the following entities:

- WORKQUEUE
- WORKQUEUESUBSCRIPTION

*Allocation Target*

The business object 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 the workflow administrator. The configuration data that is contained within this business object is stored on the following entities:

- ALLOCATIONTARGET

- ALLOCATIONTARGETITEM

**Miscellaneous**

*Rate Table*

The business object contains the configuration data for a rate table. Rates are values that can vary over time. An example of a rate would be 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:** The following special behaviour applies to a rate table business object: When using CER rule sets, certain changes to the system can cause widespread recalculations (e.g. 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, we recommend that the following object types get published in this order: Rate Table, CER Data Configuration, CER Rule Set and CER Benefit Product.

The configuration data that is contained within this business object is stored on the following entities:

- RATEHEADER
- RATEROW
- RATECOLUMN
- RATECELL

*Assessment of type Decision Assist*

The business object contains the configuration data for an assessment case of type decision assist. A decision assist assessment contains the questionnaires and rules (Decision Matrices) that will be used when running assessments for clients using Cúram Decision Assist (CDA), to calculate an outcome or rating for a client. These are grouped into determination packages. The configuration data that is contained within this business object is stored on the following entities:

- 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

### *Financial Calendar*

The 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 configuration data that is contained within this business object is stored on the following entity:

- FINANCIALCALENDAR

### *Language and Locale Mapping*

The 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 configuration data that is contained within this business object is stored on the following entity:

- LANGUAGELOCALEMAP

### *Intake Process*
The business object contains the data for an intake process configuration. An intake process is required in order for an agency to gather data for a client.

Cúram 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 configuration data that is contained within this business object is stored on the following entities:

- INTAKEPROCESSNAME
- INTAKEPROCESSCONFIGURATION
- INTAKEPROCESSROLEASSIGNMENT
- MILESTONECONFIGURATION
- PROGRAMTYPE

### *Priority, Complexity, Risk (PCR)*

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 configuration data that is contained within this business object is stored on the following entities:

- 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 configuration data that is contained within this business object is stored on the following entities:

- ADVICECONTEXTCONFIGURATION
- ADVICECONTEXTRULES
- ADVICECONTEXTKEY
- CONFIGURATIONPARAMETER

**Service Plan**

The business object contains the configuration data for a service plan. Cúram 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. The common types of service plans include Return to Work, Child Protection, and Permanency Plans. The configuration data that is contained within this business object is stored on the following entities:

- 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**

### *Intelligent Evidence Gathering (IEG) Script*

The business object contains the configuration data for an 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. The configuration data that is contained within this business object is stored on the following entities:

- IEGSCRIPTINFO
- IEGSCRIPTRELS
- APPRESOURCE

### *DataStore Schema*

The business object 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**

### *Fund*

This business object 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. The configuration data that is contained within this business object is contained within the following entities:

- PROGRAMFUND
- FUNDCONFIGURATION
- FISCALYEAR
- FUNDFISCALYEAR
- FUNDUSERROLE
- ORGOBJECTLINK

### *FundingRuleSet*

This business object 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 in which the funds should be used. The configuration data that is contained within this business object is contained within the following entity:

- FUNDEDITEMRULESLINK

# Infrastructure

**Note:** Some infrastructure business objects exhibit special behaviour as follows:

A target system is created using transported data from the source system. In the source system a primary key can be updated. If this primary key is subsequently transported, it is added as a new business object rather than an update to the existing one on the target system. Therefore both the original primary key name and the updated primary key will be 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.

## Rule Set

The business object contains the configuration data for a rule set. A rule set is the set of rules that is applied to case evidence in order to determine eligibility for a product. At least one rule set must be assigned to a product as a prerequisite to eligibility determination. The configuration data that is contained within this business object is stored on the following entities:

- RULESETINFORMATION
- RULESETLINK

## Application Property

The business object contains the configuration data for an application property. The properties table contains all the properties used when running an application. The properties are loaded to the database during the build database target and at runtime are cached from the database for use by the Application. The properties defined can be dynamic or static. The configuration data that is contained within this business object is stored on the following entities:

- PROPERTIES
- PROPDESCRIPTION

## Code Table

The business object 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. The configuration data that is contained within this business object is stored on the following entities:

- CODETABLEHEADER
- CODETABLEITEM

## Code Table Hierarchy

The business object 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. The configuration data that is contained within this business object is stored on the following entity:

- CODETABLEHIERARCHY

## Workflow

### *Workflow Business Process*

The business object contains the configuration data for a workflow process definition. A workflow process is enacted at runtime based on a workflow process definition. A workflow process definition describes the flow of a business process in terms understood by the Cúram 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. The configuration data that is contained within this business object is stored on the following entity:

- PROCESSDEFINITION

### *Workflow Process Enactment Event Configuration*

The business object 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. It is possible to start a process in response to an event being raised. This requires the setup of workflow process enactment event configuration data. The configuration specifies the process/processes to start in response to a specific event being raised. Process enactment event configuration is stored on the database and a user interface is supplied to allow the manipulation of this data. The configuration data that is contained within this business object is stored on the following entities:

- PROCENACTMENTEVT
- PROCENACTEVTDATA

### Event

The business object 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 notification of that event having occurred provided they are registered as listeners for that event. The configuration data that is contained within this business object is stored on the following entities:

- EVENTCLASS
- EVENTTYPE

### Application Resource

### *Application Resource Universal Access (UA)*

The business object contains the data for a Cúram 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. The configuration data that is contained within this business object is stored on the following entity:

- APPRESOURCE

### *Application Resource Image*

The business object 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. The configuration data that is contained within this business object is stored on the following entity:

- APPRESOURCE

### *Application Resource Properties*

The business object 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. The configuration data that is contained within this business object is stored on the following entity:

- APPRESOURCE

### *Application Resource Schema*

The business object 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. The configuration data that is contained within this business object is stored on the following entity:

- APPRESOURCE

### Application Resource XML

The business object 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. The configuration data that is contained within this business object is stored on the following entity:

- APPRESOURCE

### Microsoft Word Template

The business object contains the configuration data for a Microsoft Word template. A user can create a communication based on an Microsoft Word template and make any necessary modifications to the communication before issuing it. The configuration data that is contained within this business object is stored on the following entities:

- DOCUMENTTEMPLATE

### Extensible Stylesheet Language Template

The business object 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 which will appear on all communications generated from the template. The configuration data that is contained within this business object is stored on the following entities:

- XSLTEMPLATE
- XSLTEMPLATEINST

### Security

### Security Role

The business object 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. The configuration data that is contained within this business object is stored on the following entities:

- SECURITYROLE
- SECURITYROLEGROUP

### Security Group

The business object 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. The configuration data that is contained within this business object is stored on the following entities:

- SECURITYGROUP
- SECURITYGROUPSID

### Security Identifier

The business object 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. The configuration data that is contained within this business object is stored on the following entities:

- SECURITYIDENTIFIER
- SECURITYFIDSID

**User Interface**

*User Interface Application Configuration*

This business object 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. The configuration data that is contained within this business object is stored on the following entity:

- APPRESOURCE

*User Interface Section Configuration*

This business object 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. The configuration data that is contained within this business object is stored on the following entity:

- APPRESOURCE

*User Interface Tab Configuration*

This business object 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. The configuration data that is contained within this business object is stored on the following entity:

- 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. The configuration data that is contained within this business object is stored on the following entities:

- 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. The configuration data that is contained within this business object is stored on the following entities:

- PAGECONFIG
- USERPAGECONFIG

## Application Modules

**Verification**

The business object contains the configuration data for verification. Cúram 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, e.g., birth certificates or bank statements, or by verbal means, e.g., telephone calls. The configuration data that is contained within this business object is stored on the following entities:

- VERIFICATIONCATEGORY
- VERIFICATIONITEM
- VERIFIABLEDATAITEM
- VERIFICATIONITEMUTILIZATION
- VERIFICATIONREQUIREMENT
- DEPENDANTDATAITEM
- VERIFICATIONREQUIREMENTUSAGE

**Evidence Broker**

The business object contains the configuration data for the evidence broker. The Cúram Evidence Broker provides a flexible approach to evidence sharing. SEM 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. The configuration data that is contained within this business object is stored on the following entities:

- EVIDENCEBROKERCONFIG
- PRODUCTEVIDENCELINK
- ADMINICEVIDENCELINK

**Appeals**

Cúram Appeals configuration is incorporated into the business object the appeal configuration is associated with; a benefit product or an issue.

*Legal Action*

The business object 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. The configuration data that is contained within this business object is stored on the following entities:

- ADMINLEGALACTION
- ADMINLEGALACTIONLINK
- ADMINLEGALSTATUS
- CASEMULTILEGALSTATUSCONFIG
- LEGALSTATUSCASETYPELINK
- ADMINLEGALACTIONCATEGORY
- LEGALCASETYPELINK

**Cúram Provider Management**

*Service Offering*

The 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 configuration data that is contained within this business object is stored on the following entities:

- SERVICEOFFERING
- SERVICERATE
- PAYMENTOPTIONS
- SERVICEDELIVERYCONFIGURATION

- CREOLERULESETLINK
- SOSECURITYRIGHTS
- ABSENCEREASONCONFIGURATION
- SOATTENDANCECONFIGURATION
- SOATTENDANCEPAYMENT
- SOROSTERCONFIGURATION
- SOAPPROVALCRITERION
- CONTRACTTEXTLINK
- CONTRACTTEXT
- SOEVALUATIONCRITERION
- SOPERFORMANCEMEASURELINK
- SOTRAININGREQUIREMENT
- TRAININGSERVICEOFFERING
- SERVICEOFFERINGGROUPLINK
- LOCALIZABLETEXT
- TEXTTRANSLATION

### Attendance Payment Frequency

The business object 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. The configuration data that is contained within this business object is stored on the following entity:

- ATTENDANCEPAYMENTFREQUENCY

### Roster Generation Frequency

The business object 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. The configuration data that is contained within this business object is stored on the following entity:

- ROSTERGENERATIONFREQUENCY

### Training

The business object 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. The configuration data that is contained within this business object is stored on the following entities:

- TRAINING
- TRAININGCREDIT

### Certification

The business object 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. The configuration data that is contained within this business object is stored on the following entities:

- CERTIFICATIONCHANGEHISTORY
- CERTIFICATION
- ORGANISATIONUNIT

### Performance Measure

The business object contains the configuration data for a performance measure. A performance measure defines the criteria by which the performance of a provider can be measured. The configuration data that is contained within this business object is stored on the following entity:

- PERFORMANCEMEASURE

### Placement Payment Frequency

The business object 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. The configuration data that is contained within this business object is stored on the following entity:

- PLACEMENTPAYMENTFREQUENCY

### Service Evaluation Criterion

The business object 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. The configuration data that is contained within this business object is stored on the following entities:

- SERVICEEVALUATIONCRITERION
- RESPONSESCORE

### Absence Reason Configuration

The business object 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. The configuration data that is contained within this business object is stored on the following entity:

- ABSENCEREASONCONFIGURATION

### Service Group

The business object contains the configuration data for a service group. A service group is used to group a number of related services. The configuration data that is contained within this business object is stored on the following entities:

- SERVICEGROUP
- SERVICEOFFERINGGROUPLINK

## Cúram Universal Access

### Intake Application
The 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 forth. This information becomes evidence on the client's case and is used to determine eligibility.

The configuration data that is contained within this business object is stored on the following entities:

- INTAKEAPPLICATIONTYPE
- APPLICATIONCATEGORY
- APPLICATIONCATEGORYLINK
- PDFFORM
- WITHDRAWALREQUESTREASON
- APPLICATIONTYPEPROGRAMLINK

- PROGRAMTYPE
- EXTERNALSYSTEM
- PDFFORMPROGRAMLINK
- SERVICEAREA
- SERVICEAREALINK
- PDFFORMOFFICESERVICEAREALINK
- MAPPINGCONFIGURATION
- PROGRAMOFFICESERVICEAREALINK
- ONLINEPROGRAMGROUP
- ONLINEPROGRAMGROUPLINK
- LOCALIZABLETEXT
- TEXTTRANSLATION

### Application Case

The 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 etc. When an intake application is submitted, an application case is created. The application case stores all the information submitted with the intake application. The configuration data that is contained within this business object is stored on the following entities:

- 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 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 configuration data that is contained within this business object is stored on the following entities:

- SCREENINGTYPE
- SCREENINGRULESLINK
- SCREENINGCATEGORYLINK
- APPLICATIONCATEGORY
- LOCALIZABLETEXT
- TEXTTRANSLATION
- ONLINEPROGRAMITEM
- ONLINEPROGRAMGROUP
- ONLINEPROGRAMGROUPLINK

- PROGRAMTYPE
- MAPPINGCONFIGURATION
- PDFFORMPROGRAMLINK
- PDFFORM
- PROGRAMOFFICESERVICEAREALINK
- SERVICEAREA
- PDFFORMOFFICESERVICEAREALINK
- SERVICEAREALINK
- EXTERNALSYSTEM
- REMOTESYSTEM
- TARGETSYSTEM

### *Triage*

The 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 will quickly identify these needs and provide details of suitable services, service providers and programs to help meet those needs.The configuration data that is contained within this business object is stored on the following entity:

- TRIAGECONFIGURATION

### *Life Event*

The 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 e.g. 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 configuration data that is contained within this business object is stored on the following entities:

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

***Outreach***

The 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. The campaigns are based on information that the agency knows about the citizen. For example, if the citizen is unemployed and has a child of school going age, information on child care can be displayed, which will assist the citizen when they find work. The configuration data that is contained within this business object is stored on the following entities:

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

***Motivation***

The 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 configuration data that is contained within this business object is stored on the following entities:

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

**Cúram Social Enterprise Collaboration**

*(deprecated) Social Enterprise Folder*

> **Note:** Social Enterprise Folders (SEFs) are deprecated. For more information, see Deprecated features.

The business object 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. The configuration data that is contained within this business object is stored on the following entities:

- SOCIALENTERPRISEFOLDERADMIN
- LOCALIZABLETEXT
- TEXTTRANSLATION

### Multidisciplinary Team Member Role

The business object 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. The configuration data that is contained within this business object is stored on the following entities:

- MDTMEMBERROLE
- MDTSHARINGCONFIGRESTRICTION
- MDTSHARINGCONFIGURATION
- LOCALIZABLETEXT
- TEXTTRANSLATION

## Cúram Outcome Management

### 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 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 configuration data that is contained within this business object is stored on the following entities:

- OUTCOMEPLANACTIONADMIN
- LOCALIZABLETEXT
- TEXTTRANSLATION

### Outcome Plan Goal

The 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 configuration data that is contained within this business object is stored on the following entities:

- OUTCOMEPLANGOALADMIN
- LOCALIZABLETEXT
- TEXTTRANSLATION

### Outcome Plan Objective

The 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 configuration data that is contained within this business object is stored on the following entities:

- EXPECTEDOUTCOME
- TEXTTRANSLATION
- LOCALIZABLETEXT

### Condition Data

The 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 configuration data that is contained within this business object is stored on the following entities:

- CONDITIONDATA

### Condition Set

The 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 configuration data that is contained within this business object is stored on the following entities:

- 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 e.g. 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 i.e. an outcome plan factor can be added to an outcome plan in a standalone manner. The configuration data that is contained within this business object is stored on the following entities:

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

### Outcome Management Questionnaire

The business object contains the configuration data for the questionnaire used by an assessment definition which uses a CDA matrix to produce assessment results. The configuration data that is contained within this business object is stored on the following entities:

- QUESTION
- QUESTIONNAIRE
- QUESTIONNAIREVERSION

### Assessment

The 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 configuration data that is contained within this business object is stored on the following entities:

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

### CDA Assessment Configuration

The 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 configuration data that is contained within this business object is stored on the following entities:

- DAASSESSMENTCONFIGURATION
- DAOUTCOMEFACTORLINK
- DAOUTCOME
- DECISIONMATRIX
- DECISIONMATRIXVERSION
- DMOUTCOMELINK
- DMQUESTIONLINK

### CER Assessment Configuration

The 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 configuration data that is contained within this business object is stored on the following entities:

- CERTSPECIFICATION
- LOCALIZABLETEXT
- TEXTTRANSLATION

### *Federal Allowable Component*

The 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 configuration data that is contained within this business object is stored on the following entities:

- FEDERALALLOWABLECOMPONENT
- FEDERALALLOWABLECOMPONENTLINK
- LOCALIZABLETEXT
- TEXTTRANSLATION

## Cúram Family Services

### Contact Log

The business object 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. The configuration data that is contained within this business object is stored on the following entities:

- CONTACTCOMPLIANCECASETYPE
- CONTACTCOMPLIANCEFACTOR
- CONTACTCOMPLIANCEFREQUENCY

### Placement Type Mapping

The business object 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. The configuration data that is contained within this business object is stored on the following entity:

- PLACEMENTSERVICEOFFERINGCONFIG

### Master Detail Code Table

The business object contains the configuration data for a master detail code table. Master detail code table is used to define code table hierarchy in Cúram Family Services Suite (CFSS). 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. The configuration data that is contained within this business object is stored on the following entities:

- CODETABLECOMBOTYPE
- CODETABLECOMBOITEM

## Cúram Income Support Suite

Use this information to learn about the Cúram Income Support Suite and to understand what is contained in the following business objects.

### Summary Rules

The business object 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.

The configuration data that is contained within this business object is stored on the following entities:

- SUMMARYVIEWDEFINITION
- SUMMARYVIEWFIELDDEFINITION
- SUMMARYVIEWGROUPDEFINITION

### Benefit Delivery

The business object 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.

The configuration data that is contained within this business object is stored on the following entities:

- COUNTYDETAILS
- COUNTYDETAILSLINK
- FEEFORSERVICE
- FEEFORSERVICEPCPLINK
- PRIMARYCAREPROVIDER
- THIRDPARTYPRODUCT
- THIRDPARTYPRODUCTPCPLINK

## Summary

A summary of the main CTM concepts.

- A change set is the mechanism provided in CTM for packaging configuration data and transporting it from one system to another. It enables an organization to properly test configuration data changes before making them live on a production system.
- Transportable business objects are provided in CTM to package all the configuration data supporting a business object in a change set. This takes away the analysis required to understand the configuration data that constitutes a business object.
- CTM allows bi-directional transport of configuration data between systems in the system landscape whereby configuration data can be transported between systems based on business needs.
- CTM can be used to transport configuration data changes between systems in a structured manner leading to the modification of configuration data in a production system. It does not absolve responsibility with respect to the configuration data changes, as incorrect configuration data changes to the production system can result in application issues in the same way as without the use of CTM.
- CTM provides a feature to undo the configuration changes applied to a system should application issues be encountered after applying the configuration data in the transported change set.
- CTM 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 when the system goes into an unstable or unpredictable state through the use of CTM.

# Notices

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

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

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

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

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

*Intellectual Property Licensing Legal and Intellectual Property Law IBM Japan Ltd. 19-21, Nihonbashi-Hakozakicho, Chuo-ku Tokyo 103-8510, Japan*

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

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

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

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

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

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

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

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

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

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

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

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

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

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

COPYRIGHT LICENSE:

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

## Privacy Policy considerations

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

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

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

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

## Trademarks

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

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

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

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

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

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

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

**IBM**®

Part Number: