IBM Cúram Social Program Management

Cúram Advisor Configuration Guide

Version 6.0.4

Note

Before using this information and the product it supports, read the information in Notices at the back of this guide.

This edition applies to version 6.0.4 of IBM Cúram Social Program Management and all subsequent releases and modifications unless otherwise indicated in new editions.

Table of Contents

Chapter 1

Introduction

1.1 Purpose

The purpose of this guide is to provide information on how to configure the Cúram Advisor to display appropriate context sensitive advice to users. This guide is intended to be a technical overview of the configuration capabilities of the Cúram Advisor and the functionality it provides.

1.2 Audience

This guide is intended for administrators and developers responsible for configuring the Cúram Advisor. It is assumed that the administrators have worked with code tables and application property files as part of Cúram administration.

The rules specific configurations described in this guide are intended for rules developers responsible for developing Cúram Advisor rules. In order to understand the rule set creation chapter of this guide, it is also assumed that the developers are familiar with Cúram Express Rules (CER) and the Cúram Express Rules Editor.

1.3 Related Reading

There are several available documents that relate to the topics covered in this guide. These documents provide more detailed information on the topics covered in this guide. The following table provides a brief description of these documents:

Document Name	Description
Cúram Common Intake Guide	This document provides a detailed overview of Cúram Common Intake functionality.

Document Name	Description
Working with Cúram Express Rules	This provides step by step instructions on how to create Cúram Express Rule rule sets and how to use the Cúram Express Rules Editor to add business and technical logic to a rule set.
Cúram Express Rules Reference Manual	This provides detailed information on the Cúram Express Rules language, development environment and runtime features.

Table 1.1 Related Reading

1.4 Technical Overview

The Cúram Advisor is a configurable infrastructure which:

- Allows CER rules to be written to analyse the state of data in the database, and
- Presents the output of the analysis to users in their relevant context

To configure the Advisor, an administrator must define a new advice context as part of application administration, associate relevant application pages in the application with that context, associate pre-configured rule sets that determine the appropriate advice for that context, and configure the rule object converter and propagator. This results in the display of appropriate context sensitive advice to users at runtime.

1.5 Chapters in this Guide

Configuring an Advice Context

This chapter describes how to configure an advice context. This includes defining an advice context, associating relevant application pages and parameters with the context, and assigning one or more pre-configured rule sets to the advice context.

Creating an Advisor Rule Set

This chapter describes how to create an Advisor rule set within the CER Editor for association with an advice context. Rule set creation includes the creation of a new rule set and the creation of advice rule classes, parameters, and links. In order to appear at the case level, advice text is also specified using application property files and then linked to the rule class attributes.

Working with Evidence Rule Objects

This chapter describes the Advisor works with rule objects for active/ in-edit case evidence using a converter and propagator.

Example Advisor Configuration

This appendix provides a step-by-step example of how to configure the Advisor to display appropriate context-senstive advice.

Chapter 2

Configuring an Advice Context

2.1 Introduction

This chapter provides step by step instructions on how to configure an advice context. The advice context defines the context in which advice should appear.

Each context contains a set of application pages and page parameters that make up the context, as well as a set of rule sets which determine the advice that will appear in that context. Once the advice context is configured, the system will display appropriate contextual advice that relates to the application page the user is on.

2.2 Defining the Advice Context

To define a new advice context:

1. Login to the Cúram Administration Application.
2. In the Shortcuts Panel for the Administration Workspace, select the User Interface link.
3. Select the Advisor link to open the Advisor list page.
4. Select the New link to open the New Advisor Configuration Context dialog.
5. In the New Advisor Configuration Context dialog, enter a unique name for the advice context. The name is the logical identifier for the advice context, therefore it must be unique to other existing advice contexts.
6. Click Save.

2.3 Adding Pages to an Advice Context

Once the advice context is defined, the next step is to add the names of the application pages on which advice relating to the advice context will appear. One or more pages can be added to each advice context. To add a page to an advice context:

1. Select the New Advice Context Key menu item to open the Add Advice Key dialog.
2. In the dialog, enter the name of the page that is being added to the advice context. Note that this is the PAGE_ID from the related UIM file, e.g., Person_homePage.
3. Select Page from the Type drop-down list. Note that Page is the only available item in the drop-down list. Note that for compliancy reasons, the code table for the Advice Context Key Type is not customizable.
4. Click Save.

2.4 Adding Parameters to Pages

The parameters that relate to a page are the heart of the advice context. The content of a particular page is determined by both the page and the parameters that drive the page. For example, in order to display advice on the Person Home page, the Person_home page ID was specified when that page was added to the advice context. However, to make the advice context truly context sensitive, the page parameter for the Person Home page must also be added. The page parameter for the Person Home page is the concernRoleID. The concernRoleID page parameter enables the advice displayed at the case level to be truly context sensitive by narrowing the advice context to apply to the home page for a specific person.

To add a parameter to a page:

1. Select the New Page Parameter menu item to open the New Parameter dialog.
2. In the dialog, enter the name of the parameter, e.g., concernRoleID.
3. Select the parameter type from the Type drop-down list, e.g., number, date, string.
4. Click Save.

2.5 Assigning Rule Sets to an Advice Context

In order for the system to display context sensitive advice to the user, one or more rule sets must be assigned to each advice context. These are the rule sets that calculate the advice that relates to the pages assigned to the advice context. In order to assign a rule set to an advice context, the rule sets that are assigned to an advice context must first be created using the CER Editor.

For an overview of how to create an Advisor rule set, see Chapter 3, *Creating an Advisor Rule Set*.

To assign a created rule set to an advice context:

1. Select the Add Rule Set menu item.
2. In the dialog that appears, select the appropriate rule set from the drop-down list of available rule sets in the system.
3. Click Save.

Chapter 3

Creating an Advisor Rule Set

3.1 Introduction

This chapter provides step by step instructions on how to create an Advisor rule set. To create an Advisor rule set, a rule set must first be defined as part of application administration and then the Advisor rules classes that are contained in the rule set must then be developed using the CER Editor.

3.2 Creating a New Advisor Rule Set

To create a new rule set as part of application administration, follow the steps for creating a new rule set described in Section 2.2 of the Working With Cúram Express Rules guide. Once the rule set is created, advice rules classes that are contained in the rule set must be created using the CER Rules Editor. The following sections describe the rule classes that must be created.

3.3 Creating an Advice Context Rule Class

Each advice rule set must contain a single rule class which inherits from Abstract Advice Context. This rule class is the root rule class for the rule set and will be the rule class that produces the advice items that are displayed at the case level.

To create an advice context rule class:

1. Create the rule class using the CER Editor.
2. Give the rule class an appropriate name.
3. Specify that the rule class inherits from AbstractAdviceContext defined in the rule set named CoreAdvisorRuleSet.

3.3.1 Mandatory Rule Attribute

The following table describes the mandatory rule attribute that must be specified for each advice context rule class:

Rule Attribute Name	Attribute Type	Purpose
advice	List	This purpose of this attribute is to allow the rule set developer to specify the advice items that this rule set produces. The rule set developer must specify the derivation of this attribute to be a list of rule objects where the rule class is a subclass of abstract advice item.

Table 3.1 Mandatory Rule Attribute

3.3.2 Other Rule Attributes

Each advice context rule class should also define a field for every page parameter that will be passed in from the page. For example, if the AdviceContext is related to the PersonHome page, then the page parameter is the concernRoleID. In order to produce advice that is relevant to a specific person, the rule set must have the concern role id available to it during processing.

This is achieved by creating a rule attribute of type NumberParameter with the name of the page parameter on the advice context rule class. Provided the attribute has the same name as the page parameter, the Cúram Advisor infrastructure will automatically detect the presence of the attribute and populate it with the correct value at run time.

3.4 Creating an Advice Item Rule Class

As described previously, the purpose of each advice context is to produce a list of advice items that are associated with the context. To create an advice item for the list, the rule set developer must first create a rule class which defines the advice item, and then add an instance of the rule class to the list of advice on the advice context.

To create an advice item rule class:

1. Create the rule class using the CER Editor.
2. Give the rule class an appropriate name.
3. Specify that the rule class inherits from AbstractAdviceItem defined in the rule set named CoreAdvisorRuleSet.

The following subsections describe the rule attributes that can be specified.

3.4.1 Mandatory Rule Attributes

The following table describes the mandatory rule attributes that must be specified for each advice item rule class:

Rule Attribute Name	Attribute Type	Purpose
adviceText	String	The purpose of this attribute is to allow the rule set developer to link to the text that will appear if this advice item is triggered. Note that the value that this attribute contains is not the actual advice text. The value of this attribute is the name of the property entry that contains the advice text. For more information, see Section 3.7, *Specifying the Advice Text* .
showAdvice	Boolean	The purpose of this attribute is to allow the rule set developer to specify the conditions under which this advice item will be displayed. If the value is calculated as TRUE, the advice will be displayed, if it is calculated as FALSE, the advice will not be displayed. These rules are the main rules for the advice item, and can examine any data related to the context in order to decide whether to display the advice.
category	CodeTableItem	The purpose of this rule attribute is to allow the rule set developer to specify the category of advice that this advice belongs to. Two available categories are provided OOTB: issue and reminder.
priority	CodeTableItem	The purpose of this rule attribute is to allow the rule set developer to specify the priority of this advice item. The priority determines the order in which the advice item will appear on the list of advice displayed for the page.
status	CodeTableItem	The purpose of this rule attribute is to allow the rule set developer to indicate whether the advice has had ac-

Rule Attribute Name	Attribute Type	Purpose
		tion taken on it or not. The status of the advice can be pending or complete.

Table 3.2 Mandatory Rule Attributes

3.4.2 Optional Rule Attributes

The following table describes the optional rule attributes that can be specified for each advice item rule class:

Rule Attribute Name	Attribute Type	Purpose
caseID	Number	The purpose of this attribute is to allow the rules developer to specify if the advice item is related to a particular case. Advisor includes a special 'awareness' of cases and evidence facility. If a caseID is specified in this attribute, the advice item will be automatically linked to the case.
evidenceType	CodeTableItem	The purpose of this attribute is to allow the rules developer to specify if the advice item is related to a particular evidenceType on a particular case. It is important to note that unless a caseID has been specified, this attribute will be disregarded by Advisor processing. However, if this attribute is specified and if the caseID has also been specified, and if this advice Item is of category 'Issue' then this advice item will appear on the Issues list for the specified evidence type when viewing the case.
relatedID	Number	The purpose of this attribute is to allow the rules developer to specify if the advice item is related to a particular evidence record. Note that unless both a case and an evidence type are specified this attribute is disregarded by Advisor processing. However, if this attribute is specified and if a case and evidence type have

Rule Attribute Name	Attribute Type	Purpose
		also been specified, and if the advice Item is of type 'Issue' then this advice item will appear on the issues list for the specified evidence record when viewing the case.
expiryDateTime	DateTime	The date time upon which this advice expires. If a piece of advice is no longer applicable from a particular point in time, then this attribute is used to specify the date from which it becomes inapplicable. If this attribute is not specified, the advice is considered to be applicable until the showAdvice attribute described in Section 3.4, *Creating an Advice Item Rule Class* evaluates to false.

Table 3.3 Optional Rule Attributes

3.5 Creating a Parameter to Be Displayed in the Advice Text

To create an parameter that can be displayed in the advice text, the author must create an attribute on the AdviceItem of type StringParameter, NumberParameter or DateParameter.

The rule set developer must create a rule attribute on the advice item using the CER Editor and give the rule attribute an appropriate name.

The rule set developer must then specify that the rule attribute is of type NumberParameter, StringParameter or DateParameter as defined in the rule set named CoreAdvisorRuleSet.

3.6 Creating a Link to Be Displayed in the Advice Text

To create a link that can be displayed in the advice text, the rules developer must create a link rule class and then add an attribute to the advice item rule class that is based on the newly created link rule class.

The Cúram Advisor infrastructure will automatically process all attributes on an advice item that are based upon the AbstractLink class and convert them to links.

The rule set developer should create a rule class using the CER Editor and

give the rule class an appropriate name. Specify that the rule class inherits from AbstractLink defined in the rule set named CoreAdvisorRuleSet.

3.6.1 Mandatory Rule Attributes

The following table describes the mandatory rule attributes that must be specified for each link rule class:

Rule Attribute Name	Attribute Type	Purpose
target	String	The target of the link. If this is an internal link, the target should be the PAGE_ID of the UIM page that this link is to. If this is an external link, the target should be a fully qualified URL.
name	String	The name of the link. This name is a logical name used to reference the link in the advice text.
modal	Boolean	If the derivation of this attribute is set to TRUE, the link will appear in a modal. If the derivation of this attribute is set to FALSE, the link will appear in the main content pane.
external	Boolean	If the derivation of this attribute is set to TRUE, the link will be interpreted as a link to an external website outside of the application. If the derivation of this link is set to FALSE, the link will be interpreted as an internal link, i.e., a link to a UIM page.

Table 3.4 Mandatory Rule Attributes

3.7 Specifying the Advice Text

Advice text is specified by creating an entry in a localized property file in the Application Resources area of Application Administration. To specify the advice text:

1. Login to Application Administration.
2. Within the Shortcuts area of the Administration Workspace area, select Intelligent Evidence Gathering.
3. Select Application Resources and create a new resource which is a

property called AdviceContext.<AdviceContextName> where AdviceContextName is the name of the Advice Context as defined in Application Administration. For example, AdviceContext.PersonHomeAdvisor .

4. Create a property entry in the property file <adviceText> where adviceText is the value of the attribute adviceText on the AdviceItem rule class described in Table 3.2. The value of this property is the localized text to be displayed in the application.

Chapter 4

Working with Evidence Rule Objects

4.1 Introduction

When the Advisor is run, the advice text specified in Section 3.7, *Specifying the Advice Text* , is displayed based on an output of the advice rules run against configured evidence types. The design for advice often requires access to the latest evidence data for a case regardless of whether that evidence has yet been activated. Access to the latest active/in-edit evidence is provided by the Active/In-Edit Succession Set Rule Object Converter, which converts active/in-edit evidence data into rule objects for use by CER.

When evidence data changes, the corresponding rule object propagator (Active/In-Edit Succession Set Rule Object Propagator) notifies the Dependency Manager of the change in evidence so that previously-calculated advice can be marked as requiring recalculation.

4.2 Active/In-Edit Succession Set Rule Objects

The Active/In-Edit Succession Set Rule Object Converter is responsible for automating the conversion of a succession set of evidence to a rule object, regardless of whether the succession set contains active evidence records or the evidence records that are in edit.

Important

The Active/In-Edit Succession Set Rule Object Converter should not be confused with the Active Succession Set Rule Object Converter, nor the Active/In-Edit Succession Set Rule Object ropagator confused with the Active Succession Set Rule Object Propagator.

However, the active in/edit succession set rule objects share important characteristics with active succession set, including the population of timeline/non-timeline data and relationships to other rule ob-

jects. See the Inside Cúram Eligibility and Entitlement Using Cúram Express Rules guide for details on processing for active succession set rule objects.

Temporal evidence functionality allows developers to store records of evidence that can change over time. When circumstances change in the real world, a user can record those changes in the system by "succeeding" an earlier version of evidence. These versions of evidence make up a "succession set" which describe the history of some real-world evidence. Advisor rules treat each piece of changeable evidence as one rule object which has timeline-based attributes.

The Active/In-Edit Succession Set Rule Object Converter automates the conversion of a succession set of evidence rows into a single rule object with a mixture of timeline and non-timeline attributes. Changes to evidence go through a lifecycle. Transitions between the states of the evidence lifecycle are considered by the Active/In-Edit Succession Set Rule Object Propagator.

The Active/In-Edit Succession Set Rule Object Converter also populates relationships between rule objects for parent and child succession sets, if required.

4.3 Configuration

The converter and propagator share a common set of configuration data, and accept configurations which adhere to the following structure:

- propagator type must be "ROPT2011" (the code for 'Active/InEdit Succession Set Propagator' from the 'RuleObjectPropagatorType' code table);
- each evidence type to be converted or propagated must be listed in an 'evidence' element with a type exactly matching the evidence's type from the 'EvidenceType' code table; and
- each conversion/propagation target must be listed as a 'ruleset' element (within the 'evidence' element), specifying the name of the rule set to target, and optionally the rule class (if the name of the rule class differs from that of the database table).

  Configurations are cumulative, i.e., there may be many configurations of type "ROPT2011", and if an evidence type is present in any of those configurations then the evidence type will be converted and propagated; otherwise, the evidence type will be ignored.

```
<?xml version="1.0" encoding="UTF-8"?>
<!--Code for the 'Active/InEdit Succession Set  Propagator" from
    the 'RuleObjectPropagatorType' code table./>
-->
<propagator type="ROPT2011">
  <configuration>
    <!--
    The code for your evidence type.
```

```
    The type must exactly match the code from the
    'EvidenceType' code table including upper/lower case.
    -->
    <evidence type="ET10069">
      <ruleset name="TestActiveInEditPropagatorRuleSet"
ruleclass="SimpleTestActiveInEditPropagatorRuleSet"/>
      <!--
      Map this evidence type to
      TestActiveInEditPropagatorRuleSet.
      SimpleTestActiveInEditPropagatorRuleSetCalculatedDates,
      and also to CGISSAdvisorEvidenceRuleSet.LivingArrange
      -->
      <ruleset name="TestActiveInEditPropagatorRuleSet"
ruleclass=
"SimpleTestActiveInEditPropagatorRuleSetCalculatedDates"/>
      <ruleset name="CGISSAdvisorEvidenceRuleSet"
            ruleclass="LivingArrange"/>
    </evidence>
    <evidence type="ET10090">
      <ruleset name="CGISSAdvisorEvidenceRuleSet"
        ruleclass="HouseholdRelationship"/>
    </evidence>
    <evidence type="ET10087">
      <ruleset name="CGISSAdvisorEvidenceRuleSet"
        ruleclass="Deprivation"/>
    </evidence>
  </configuration>
</propagator>
```

Example 4.1 Sample configuration for the Active/In-Edit Succession Set Rule Object Converter and Propagator

The following types of configuration problems will be detected by the Active/In-Edit Succession Set Rule Object Converter/Propagator processing:

- Evidence type not specified in the `evidence` element;
- The evidence type with the specified type code could not be found;
- The targeted rule class does not extend the `PropagatorRuleSet.ActiveInEditSuccessionSet` rule class; and
- A rule class is targeted by more than one source evidence type.

4.4 Conversion Processing

Each evidence type may map to a number of target rule classes, according to the configurations for the Active/In-Edit Succession Set Rule Object Converter held on the system. However, for the sake of clarity, the rest of this section describes the behavior of the Active/In-Edit Succession Set Rule Object Converter in the situation where an evidence type is mapped to a single rule class only.

When an Active/In-Edit Succession Set Rule Object is requested during a CER calcuation, the Active/In-Edit Succession Set Rule Object Converter is invoked to populate that rule object. The Active/In-Edit Succession Set Rule Object Converter will retrieve the latest evidence rows in the succession set and use them to populate the attribute values on the rule object.

The values of the evidence fields are used to map to identically-named rule attributes on the rule class. Any evidence field without a corresponding rule attribute is ignored. Evidence fields are defined by:

- **Dynamic evidence.** The evidence fields available are those defined by the dynamic evidence metadata for the evidence type (see the Cúram Dynamic Evidence Configuration Guide); and
- **Non-dynamic evidence.** The evidence fields available are those defined on the evidence-specific database table modeled for the static evidence type (see the Cúram Temporal Evidence Guide and the Cúram Evidence Generator guides).

When populating a particular attribute value on a rule object, the behavior of the Active/In-Edit Succession Set Rule Object Converter differs according to whether the rule attribute's type is a Timeline. The Active/In-Edit Succession Set Rule Object Converter also contains special processing to populate rule attributes to point to rule objects for related succession sets.

4.4.1 Timeline Based Data Types

If the data type of the attribute is Timeline<some data type>, then the Active/In-Edit Succession Set Rule Object Converter allows the possibility of the evidence value to differ (across different evidence versions in the succession set). The converter will form a timeline value with the start date equal to the start of the evidence lifetime, end date equal to the end of the evidence lifetime, and accumulated varying values over the lifetime of the evidence

4.4.2 Non Timeline Based Data Types

If the data type of the attribute is not Timeline<some data type>, then the converter does not allow the possibility of the evidence value to differ across different evidence versions in the succession set. Ordinarily each version of evidence in the succession set should bear the same data value for the evidence field, and this single data value will be used to populate the rule attribute value.

4.4.3 Population of Relationships to Rule Objects for Other Succession Sets

When a succession set of active evidence is converted to a rule object, then any rule attributes which are annotated with the 'relatedSuccessionSet' will be automatically populated with rule objects for related succession sets:

- parent: the attribute will be populated with the rule objects for the succession sets for the parent(s) of the evidence; or
- child: the attribute will be populated with the rule objects for the succes-

sion sets for the child(ren) of the evidence.

The type of the related evidence is identified from the type of the attribute, which can either be a rule class (extending 'ActiveInEditSuccessionSet') or a list of such rule classes.

4.4.4 Rule Attributes Inherited from ActiveInEditSuccessionSet

Each rule class targeted by the Active/In-Edit Succession Set Rule Object Propagator must ultimately extend the 'PropagatorRuleSet.ActiveInEditSuccessionSet' rule class, and so will inherit the following rule attributes:

- successionID:

  Populated from the 'successionID' value on the "EvidenceDescriptor" rows, and used to uniquely identify the rule object (amongst other rule objects of the same rule class);

- caseID:

  Populated from the 'caseID' value on the 'EvidenceDescriptor' row. If the evidence relates to an integrated case, the case ID will be that of an integrated case; if the evidence relates to a product delivery case, the case ID will be that of the particular product delivery that holds the evidence;

- description:

  Contains a default rule to derive a description for the succession set rule object; sub-classes are free to override this description if required;

- exists:

  A Boolean timeline which indicates the period of time for which the succession set rule object "exists", i.e. true for the dates between the designated start and end dates (inclusive), and false for dates before the start of the lifetime or after its end, if any; and

- evidenceDescriptorID:

  A Number timeline, populated from the "evidenceDescriptorID" value on the "EvidenceDescriptor" rows which make up the succession set. The values vary according to the evidence row "in effect" at various points along the lifetime of the succession set rule object. Each value uniquely identifies the active "EvidenceDescriptor" row which contains the source of the data in effect on a particular date on the timeline-based attributes on the rule object. Note that these values will change when an evidence correction is activated, because at that point a different evidence row becomes an active member of the succession set.

Restrictions on Access

In your CER rule sets you will use CER's `<readall>/<match>` expression to access rule objects converted from active/in-edit succession set data.

You may only specify the `retrievedattribute` to be the `caseID` .

If you attempt to specify a `retrievedattribute` to be the name of any other attribute, then the Active/In-Edit Succession Set Rule Object Converter will throw a runtime exception when the CER `<readall>/<match>` expression is executed.

Tip

If you require only some of the active evidence row evidence of a given type for a case, then consider wrapping the `<readall>/<match>` expression within a `<filter>` expression to return only the data you require, e.g. use `<readall>/<match>` matching on `caseID` to find all the `Income` active/in-edit succession set rule objects for a case, and then use a `<filter>` to restrict the rule objects to just those for a particular member of the case.

You may specify the `ruleset` and `ruleclass` for the `<readall>` expression to be a rule class mapped by the data configuration. If you attempt to specify a rule class which is not directly mapped (e.g. a base rule class that you have created from which your concrete rule classes inherit) then no rule objects will be found.

Caution

Do not use a `<readall>` without a `<match>` .

Such an unqualified `<readall>` would typically retrieve a very large number of rule objects and no dependency on the overall set of rule objects will be stored.

Precedents Identified

If Active/In-Edit Succession Set Rule Objects are accessed during a CER calculation, and the CER utility is used to identify precedents, then the following precedents will be identified:

Name	When Identified	Trigger for Recalculation
Active/In-Edit Evidence	Identifies any case for which [1] : • a search was executed to retrieve Active/In-Edit Succession Set Rule Objects; and/or • one or more attribute values were accessed for one or more Active/In-Edit	If active evidence is edited or canceled, or changes made to in-edit evidence for a case, then a precedent change item for the case will be written to a precedent change set [2] .

Name	When Identified	Trigger for Recalculation
	Succession Set Rule Objects for the case's evidence The precedent ID refers to the caseID which owns the evidence that was accessed.	
Rule Object Data Configurations	Identifies the use of the configuration for the Active/In-Edit Succession Set Rule Object Converter if any Active/In-Edit Succession Set Rule Object is accessed during the calculation.	If changes to the data configuration for the Active/In-Edit Succession Set Rule Object Converter are published, then a precedent change item for the converter's data configuration will be written to a precedent change set.

Table 4.1 Precedents Identified for Active/In-Edit Succession Set Rule Objects

4.5 Propagation Processing

When evidence edits are made for an evidence type that is configured for Active/In-Edit Succession Set Rule Objects, then the Active/In-Edit Succession Set Rule Object Propagator listens to internal events from the Evidence Controller, requests the corresponding rule object and manipulates it in memory.

A rule object may be created, modified or removed, according to whether evidence is being created, edited or canceled (in the evidence workspace, prior to activation).

The Active/In-Edit Succession Set Rule Object Propagator informs the Dependency Manager of evidence edits that have been made so that the Dependency Manager can determine the effects of those changes, in particular to mark advice as requiring recalculation. Dependencies on active/in-edit evidence are stored at the case level, by recording a dependency on the caseID of the case that owns the evidence.

4.6 Example

Let's say that a person's income from an employment is modeled as Cúram Temporal Evidence. The income starts when a person starts an employment, and ends if the employment is subsequently terminated. The name of the employer is constant throughout the period of income, because the designer of the evidence structure made a design decision that if a person moves from one job to another, then the first employment comes to an end and a separate

employment starts.

Over the lifetime of an employment, the income amount (i.e. the per annum pay) can vary, as the employee receives pay rises. Similarly, but independently, the person can be employed on a permanent or temporary basis, and this "employment status" can change over the lifetime of the employment. It is possible for the income's amount to change on the same date as the employment status, but a change in income amount can occur without a change in employment status, and vice versa.

The evidence designer designs an Income evidence entity as follows:

- startDate:

  The date that the income (i.e. the overall employment) started;

- endDate:

  The date that the income (i.e. the overall employment) ended, if any;

- employer; and

  Identifier of the employer, constant throughout the income (see the design decision described above);

- amount; and

  The per-annum pay amount;

- employmentStatus:

  Code for whether the employment status is permanent or temporary

A rules designer then models a new "Income" rule class, extending the "ActiveInEditSuccessionSet" rule class, and adds rule attributes, identifying which have values that change over time (i.e. those which should be allowed to vary across different records in the same succession set):

Should be constant across records in the succession set:

- startDate;
- endDate;
- employer; and

Should be allowed to vary across records in the succession set:

- amount; and
- employmentStatus

The rules designer also identifies which rule attributes identify the "lifetime" of the Income:

- startDate; and
- endDate

and annotates the rule class to identify these rule attributes.

An administrator publishes the rule set changes, and then publishes a data configuration for Active/In-Edit Succession Set Rule Objects to map the Income evidence type to the new rule class. A case worker records some new Income evidence (for an employment which started on 1st January 2000).

Initially the evidence is "in edit" and its data is available to the Active/ In-Edit Succession Set Rule Object Converter to populate a rule object. When evidence capture is complete, the case worker activates the evidence. No action is taken by the Active/In-Edit Succession Set Rule Object Propagator.

Over time, real-world circumstances change:

- on 1st January 2001, the income amount increases; and
- on 1st May 2002, the employment status changes from "temporary" to "permanent"

On each of these occasions, the case worker records a new version of the Income evidence, leading to the system storing a new 'Evidence-Descriptor'/'Income'pair of rows for the evidence data effective from each change date.

The Active/In-Edit Succession Set Rule Object Converter recognises that the three versions of evidence relate to a single succession set, and use the effective-dated data to change the timeline values for the attributes on the single rule object. The rule object data will be updated as soon as the changes are made, there will be no wait until the succession set is activated.

On 30th June 2002, the employment comes to an end and a case worker records the end date on the latest record in the succession set. The case worker inserts the changes, which cause the existing latest "Evidence-Descriptor"/"Income" pair to become "superseded" and a new pair to become "active". The Active/In-Edit Succession Set Rule Object Converter immediately updates the rule object to change its timeline values from 1st July 2002 (the day after the end of the employment).

Some time later, a review of the case finds that the entire history of the income has been recorded against the wrong person. All the evidence records for the Income are canceled by the caseworker, pending removal, which causes existing rule object to be removed. The case worker than realizes that he has canceled the Income record for the wrong person. He reverts the changes and the appropriate rule object gets re-created. The case worker now cancels the Income records for the correct person.

The evidence is re-recorded against the correct person (in a new succession set) and a new rule object created for the new succession set of income records. Note that the old rule object is removed and a new created before any of the evidence changes has been activated. The caseworker eventually activates the changes, causing no updates for the existing rule objects.

Notes

[1]In practice these two conditions amount to the same thing - that Active/ In-Edit Succession Set Rule Objects for the case's evidence were accessed in some way. Generally, a search will be executed to retrieve rule objects in order that one or more attribute values can be accessed on those rule objects anyway.

[2] Note though that the *activation* of in-edit evidence changes for a case does not cause a precedent change item to be written - as the newly-activated evidence is still the latest active/in-edit evidence for the case.

Appendix A

Example Advisor Configuration

A.1 Introduction

This appendix provides an example of how to configure the Cúram Advisor to display some advice from scratch. Once these steps have been completed, the following example advice will be displayed on the Person Home page: "Capture the spouse relationship of this person". See Section A.5, *View the Advice* for an example of how the configured advice will appear in the application.

The example walks you through the following tasks: creating an Advisor rule set using the CER Editor, configuring the advice context, and specifying the advice text. Once these tasks have been completed, the configured advice will be viewable at runtime.

In order to fully understand the CER Editor-related section of this appendix, you must be familiar with the CER Editor. If you are not already familiar with the basic concepts of the CER Editor, it is recommended that you read the `Working with Curam Express Rules` guide and the `Cúram Express Rules Reference Manual`.

A.2 Create an Advisor Rule Set Using the CER Editor

The first task is to create a rule set using the CER Editor that determines whether or not the advice is displayed. Note that the example advice will be displayed if, a person's marital status is "married" and no relationship of type "spouse" has been entered. Therefore, we must configure a rule set that first looks at a person's marital status and then looks at the person's relationships in order to evaluate whether or not to display the advice. We start by creating the new rule set:

1. Login to Application Administration.
2. Within the Shortcuts area of the Administration Workspace area, select

Rules and Evidence.

3. Select the Cúram Express Rule Sets link.
4. From the Action menu, select New to create a new rule set.
5. Name the rule set SpouseRuleSet and click the Save button. Assign the rule set to a category if appropriate.
6. In the Cúram Express Rule Sets tab, search for the newly created SpouseRuleSet.
7. In the Actions menu for the SpouseRuleSet , select Continue Editing. This will open a new tab link containing the CER Editor.

We now need to design the logic for the SpouseRuleSet. From a business perspective, the advice is provided on the Person Home page under the condition that this person is married and no spousal relationships have been recorded for the person. As this is a simple logic, we can jump directly to the CER Editor to design the rules implementation.

The implementation works as follows. First we determine the person for which the advice is displayed using the concernRoleID parameter from the Person_homePage. The concernRoleID is received from the Advice Context. We then search for the person and his or her associated relationships using the concernRoleID. Finally, by checking if the person's marital status is equal to married and comparing the number of spouse relationships with '0', we can determine if the advice should be provided. To design this implementation, we complete the following steps in the CER Editor:

1. From the Rule Class menu item, create new class called SpouseAdvice.
2. In the Properties tab, click the Edit link beside the Extends field.
3. Use the Change link to let the SpouseAdvice rule class extend AbstractAdviceContext within the CoreAdvisorRuleSet.
4. From the Attribute menu item, create the following new attributes for the SpouseAdviceRuleClass: concernRoleID, adviceContextID, advice, and description in the SpouseAdvice rule class.
5. Set the Data Type field for each new attribute. For the concernRoleID attribute, set the Data Type field to be NumberParameter from the CoreAdvisorRuleSet. For the adviceContextID attribute, set the Data Type field to be Number. For the advice attribute, set the Data Type field to be List of AbstactAdviceItem in the CoreAdvisorRuleSet. For the description attribute, set the Data Type field to be Message. From the Rule Class menu item, create a new rule class called SpouseAdviceItem. Let the SpouseAdviceItem rule class extend from AbstractAdviceItem within the CoreAdvisorRuleSet.
6. From the Attribute menu item, create the following new attributes for the SpouseAdviceItem rule class: adviceText, showAdvice, priority,

status, category, adviceContext, description, and ConcernRoleID.

7. Set the Data Type field for each new attribute. For the adviceText attribute, set the Data Type to be String. For the showAdvice attribute, set the Data Type field to be Boolean. For the priority attribute, set the Data Type field to be the AdvicePriority code table. For the status attribute, set the Data Type field to be the AdviceStatus code table. For the category attribute, set the Data Type field to be the AdviceCategory code table. For the adviceContext attribute, set the Data Type field to be Number. For the description attribute, set the Data Type field to be Message. For the concernRoleID attribute, set the Data Type field to be Number.
8. Drag a Code Table operator onto the priority attribute. Within the Properties tab of the operator, set the Table to be AdvicePriority and set the value to be AP2001.
9. Drag a Code Table operator on the status attribute. Within the Properties tab of the operator, set the Table to be AdviceStatus and set the value to be AS2002.
10. Drag a Code Table operator on the category attribute. Within the Properties tab of the operator, set the Table to be AdviceCategory and set the value to be AC2001.

Now we have extended the essential attributes and we can go on to add the logic to the rules. First we must determine if the person is married or not as follows:

1. Create a new Person attribute for the SpouseAdviceItem rule class. For the Person attribute, set the Data Type field to be Person from the ruleset ParticipantEntitiesRuleSet.
2. Drag a Search operator onto the Person attribute. Double click the Search operator to open the Edit Search dialog, enter the Data Type Person previously mentioned.
3. Add a New Match to the Search operator and then drag a reference to it that references the concernRoleID attribute within the SpouseAdviceItem rule class.
4. Check the Single Item checkbox in the Properties tab of the Search operator.
5. Create a new attribute for the SpouseAdviceItem rule class and call it isMarried.
6. Set the Data Type field for the new attribute to be Boolean.
7. Drag a Compare operator to the new attribute. Edit its expression to compare the maritalStatus within the Person attribute to the Codetable MaritalStatus with the value MS2.

Now we calculate how many spouse relationships the person has as follows:

1. Create a new attribute for the SpouseAdviceItem rule class and call it relationships. For the relationships attribute, set the Data Type field to be a List of type ConcernRoleRelationship from the rule set ParticipantEntitiesRuleSet.
2. Drag a Search operator onto the Relationships attribute. Double click the attribute to open the Edit Search dialog,enter the Data Type ConcernRoleRelationship just mentioned.
3. Add a New Match to the Search operator and drag a reference to it that references the concernRoleID attribute within the SpouseAdviceItem rule class.
4. Create a new attribute for the SpouseAdviceItem rule class and call it numOfSpouseRelationships with the Data Type Number.
5. Drag a Size operator onto the new attribute. Then drag a Filter operator inside the Size operator.
6. Within the Filter operator, for the Empty List element, let this refer to the relationships attribute. For the Empty Member element, drag a new Compare operator onto it that compares the relationshipType in within the relationships attribute with the Codetable RelationshipTypeCode value RT6.

Finally, we can determine the result of the previous steps as follows:

1. Open the showAdvice attribute within the SpouseAdviceItem rule class.
2. Drag the AND operator onto it, making the isMarried the first condition. For the second condition, add a Compare operator beside the isMarried condition that compares if the numOfSpouseRelationship is equal to 0.
3. Open the adviceText attribute within the SpouseAdviceItem rule class.
4. Drag a String properties operator onto the attribute and within the Properties Tab, give it a value of startCapture.
5. Open the advice attribute within the SpouseAdvice rule class.
6. Drag a Fixed List operator onto the attribute, and fill the Data Type of this operator to be AbstractAdviceItem.
7. Drag a Create operator onto the Fixed List operator, and then create the SpouseAdviceItem rule class with parameters. The parameter concernRoleID within SpouseAdviceItem must refer to concernRoleID in SpouseAdvice. The parameter adviceContext within SpouseAdviceItem must refer to adviceContextID in SpouseAdvice.
8. Save and validate the rule set within the CER Editor.
9. If no errors occur on validation, publish the rule set in application administration using the Publish option on the Cúram Express Rule Sets

tab.

A.3 Configure the Advice Context

The next task is to configure the advice context in the dynamic environment and assign the newly created rule set to it:

1. In the Shortcuts panel of the Administration Workspace, select User Interface.
2. Select the Advisor link to open the Advisor list page.
3. Configure a new advice context and name it SpouseRules.
4. Configure a new advice context key for the newly created advice context. Set the name of the advice context key to be Person_homePage and set the Type to be Page.
5. Select the Add Rule Set link the on the Actions menu. Select the SpouseRuleSet from the drop-down list to assign the rule set to the Spouse Rules advice context.
6. In the Actions menu, add the Page Parameter under Person_homePage. The Parameter Name is set to concernRoleID and Type is set to Number.

A.4 Specify the Advice Text

Finally, we specify the advice text to be displayed as follows:

1. In the Shortcuts panel of the Administration Workspace, select Intelligent Evidence Gathering.
2. Select the Application Resources link to open the Application Resources list page.
3. Create a text file and give it an appropriate name, e.g., SpouseRules.properties.
4. Add the property entry "AdviceItem.startCapture=Capture the spouse relationship of this person." to this file.
5. Select the Add Resource button. In the dialog that opens, give the new resource a name of AdviceContext.SpouseRules and ensure that the Content Type is text/plain.
6. Upload the newly created text file.

A.5 View the Advice

If the Cúram Advisor component has been installed in the application, the advice will now be displayed in the Smart Panel on the Person Home page. The following advice is displayed if the person is married and no spouse relationships have been recorded: "Capture the spouse relationship of this person".

Notices

This information was developed for products and services offered in the U.S.A. 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

Armonk, NY 10504-1785

U.S.A.

For license inquiries regarding double-byte (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.

1623-14, Shimotsuruma, Yamato-shi

Kanagawa 242-8502 Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: 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 states 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 typograph-

ical 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 Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply 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 Corporation

Dept F6, Bldg 1

294 Route 100

Somers NY 10589-3216

U.S.A.

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.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

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-IBM products. Questions on the capabilities of non-IBM products

should be addressed to the suppliers of those products.

All 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 the names and addresses used by an actual business enterprise 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.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs.

© Copyright IBM Corp. _enter the year or years_. All rights reserved.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Programming Interface Information

This publication documents intended programming interfaces that allow the customer to write programs to obtain the services of IBM Cúram Social Pogram Management.

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/us/en/copytrade.shtml .

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