



Cúram 8.1.2

How to Build a Product Guide

Note

Before using this information and the product it supports, read the information in [Notices on page 39](#)

Edition

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

© Merative US L.P. 2012, 2024

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

Contents

Note.....	iii
Edition.....	v
1 How to Build a Product.....	9
1.1 Overview.....	9
Related Reading.....	9
1.2 Before You Start.....	10
Defining Product Requirements.....	11
What Purpose Does the Product Serve?.....	11
Who Are the Recipients of the Product?.....	11
What Rules (and Calculations) Are Required to Achieve the Purpose?.....	12
What Information Needs to be Captured for these Rules and How?.....	12
Capturing the Evidence That Feeds Into the Rules.....	13
What Results Need to Be Conveyed and How?.....	13
Securing the Product.....	13
How Will Sensitive Data Be Protected?.....	14
What Impact Do Product Changes Have on Existing Cases?.....	14
Should Only Certain Users Approve Product Deliveries?.....	14
Will Evidence Be Shared across Products?.....	15
Testing the Product.....	15
Testing Rules.....	15
Testing Results.....	15
Testing Rule Object Data Configurations.....	16
1.3 Building a Simple Product.....	16
Simple Product Requirements.....	17
Purpose.....	17
Rules Required to Achieve Purpose.....	17
Information Required by these Rules.....	17
Results to Be Conveyed.....	17
Essential to Product Building - Use What You Already Have.....	17
Creating a Simple Product Evidence Entity.....	18
How to Create the Simple Product Evidence Type.....	18
Linking the Simple Product Evidence Type to Rules.....	19
Using and Extending Rule Classes Included with the Application.....	19
Extending the DefaultProduct Rule Class.....	21
Extending the ActiveSuccessionSet Rule Class.....	21
Extending the DefaultCase Rule Class.....	22
Deriving the Eligibility and Entitlement Timelines.....	22
Configuring Rule Object Converters Included with the Application.....	23

Configuring the ActiveSuccessionSet Rule Object Converter.....	23
Configuring the Product in the Administration Workspace.....	24
Creating the Product.....	24
Associating Integrated Case Categories.....	24
Defining the Product Home Pages.....	24
Defining the Eligible Participant Types.....	25
Rules Configuration.....	25
Defining Display Categories.....	25
Defining Product Periods and Associated Rules.....	25
Defining Eligibility Determination Settings.....	26
Configuring Financial and Reassessment Processing.....	27
Associating Evidence Types with the Product or Integrated Case.....	27
Testing the Product.....	27
Creating Test Classes for Rules.....	27
Testing Rule Set Coverage Along the Way.....	27
Use Logging to Test Rule Object Data Configurations.....	28
Testing the Determination Results.....	28
1.4 Building Decision Details Rules.....	28
Basic Steps to Follow in Designing Decision Details Rules.....	29
Defining The Display Category.....	29
Defining The Display Page.....	30
Identifying the Input for the Display Rules.....	30
Extending the Decision Details Rule Set.....	30
Mapping Business Object Data to Decision Details Rules.....	31
Building & Configuring Dynamic UIM Pages.....	31
Associating Decision Details Rules to the Product Period.....	33
Testing Decision Details Rules.....	33
Additional Samples for Creating Decision Details Pages.....	33
Display Data Samples.....	34
Main Category Display Class Sample.....	35
Sample of Standard Rule Class Used by the Main Display Class.....	35
Subscreen Rule Class Sample.....	36
Example of Display Rules Derived From Evidence.....	37
1.5 (deprecated) Key Decision Factors.....	37
(deprecated) Creating Key Decision Factor Rules.....	37
(deprecated) Associating Key Decision Factor Rules to the Product Period.....	38
(deprecated) Testing Key Decision Factor Rules.....	38
Notices.....	39
Privacy policy.....	40
Trademarks.....	40

1 How to Build a Product

Use this information to build a simple product. A set of examples cover different aspects of product design. These examples show how the various components of the product are used in eligibility and entitlement processing.

1.1 Overview

The purpose of this guide is to provide a holistic view on how to build products using the Cúram Platform. It achieves this purpose by providing a set of examples, each one ranging in complexity, and covering different aspects of product design. The goal of this approach is to cover as much ground as possible on the options available to product designers. In order to accommodate a wider range of product options, this guide does not always cover the lower level, specific details. In these instances, the guide makes specific references to available documentation.

This document is intended for designers and developers who build or customize the products.

Related Reading

There are several aspects to product building that warrant their own complete set of documentation, given their extensiveness. The following table provides a detailed description of the range of documentation that is related to the overall task of building a product.

Table 1: Description of Related Documents

Document	Document Type	Relationship to Building a Product
Integrated Case Management Guide	Business Guide	This is a good starting point for understanding how products are used to define case processing.
Integrated Case Management Configuration Guide	Configuration Guide	This follows on from the business guide and provides detailed explanations on the case configuration options available in the administration component of the application, including but not limited to detailed explanations on product configuration.
Cúram Participant Guide	Business Guide	This defines the basic concepts of participants and participant types.
Cúram Person and Prospect Person Evidence Developers Guide	Developer Guide	This provides a high level technical understanding of person/prospect person evidence and its components. It also outlines the available customization options and extension points.

Document	Document Type	Relationship to Building a Product
Inside Cúram Eligibility and Entitlement Using Cúram Express Rules	Developer Guide	This is a reading prerequisite for building a product. It provides a detailed technical overview of case eligibility and entitlement processing.
Cúram Evidence Guide	Business Guide	This is a useful for understanding the basic way in which evidence is maintained at the case level.
Cúram Dynamic Evidence Configuration Guide	Configuration Guide	This provides detailed explanations on how to configure dynamic evidence, including how to create new evidence types.
Working with Cúram Express Rules	Developer Guide	This provides detailed explanations on how to create rules using the dynamic rules editor.
Cúram Express Rules Reference Guide	Developer Guide	This describes all aspects of the Cúram Express Rules (CER) language and is an essential reference to rules design.
Getting Started with Cúram Client Development Guide	Developer Guide	The aspects of this guide covering dynamic User Interface meta-data (UIM) design are of particular importance to building a product. UIM pages are required for case and case member home pages, evidence pages, and decision results pages.
Cúram Web Client Reference Guide	Developer Guide	This describes all aspects of the web client and is an essential reference to Cúram client development.

1.2 Before You Start

The task of building a new product can seem daunting - quite a bit of thought and practical work needs to go into product design and delivery. Understanding fundamental aspects of the product design before taking on the physical task of building the product can greatly simplify the complexity of the task at hand.

This section describes fundamental aspects of product design that need to be considered before building any type of product. These include the very basics of product requirement, product security requirements, testing approaches, and performance considerations.

Defining the product requirements is the most obvious task that must take place before building a product. In addition to defining product requirements up front, it is also essential to think about ahead of time how that product will be secured, how it will be tested during the building process, and ways in which the product can be made more performant. These are architectural concepts that will save a lot of time and effort if thought about up front, rather than down the line in product development.

Defining Product Requirements

At a very basic level, a product involves the delivery of a benefit or a service. A starting point for defining product requirements is to figure out the purpose of the product and its recipients. The purpose is primarily about identifying what benefits and services are meant to be delivered and how these are meant to help the product recipients.

The next level of defining product requirements deals with defining how eligibility and entitlement for the benefits and services can be effectively determined. This includes identifying the rules for determination, the information that must be captured about potential product recipients. This information is compared to the rules in order to make decisions.

A third level to product requirements is to think about how these decisions will be conveyed. Just as essential to determining eligibility and entitlement for benefits and services is the ability to understand how a person's circumstances have resulted in these eligibility and entitlement results. Without this insight, it is very difficult to see how these products meet the needs of its recipients.

In summary, the overall aim of delivery a product is to come up with a view over the lifetime of the case that will contain the "three Es" (Eligibility, Entitlement, Explanation). This concept is discussed in more detail in the *Inside Eligibility and Entitlement Using Cúram Rules Guide*.

What Purpose Does the Product Serve?

The purpose of a product can seem fairly straightforward, for example, to provide income assistance to families in need. When building a product, one quickly realizes that even the most straightforward goals are complex in their delivery. Taking the example of providing income assistance to families in need, there are three distinct parts to this purpose.

The product must:

- Identify the members of the family
- Identify the needs of the family
- Determine how much income assistance is required to meet these needs

Who Are the Recipients of the Product?

On the surface, defining the recipients of the product seems easy enough - these are the persons who will be deemed eligible. But the reality is there is quite a bit more to think about in relation to product recipients.

For example, many products are not for the benefit of one person, but for the benefit of a family, and these families can have a wide range of needs. A product that provides income assistance to families needs to take into account varying degrees of need. Families with small children may need additional assistance. The same can be said for families with members who have disabilities.

The relationships between family members can have a significant impact on a family's changing needs, for example, if the main wage earner leaves the household. Before building a product, it is essential to think about the complexities surrounding whose needs are being met in the delivery of that product.

What Rules (and Calculations) Are Required to Achieve the Purpose?

When it comes to building products, one of the more complex tasks can be defining the rules and calculations required to make decisions regarding eligibility and entitlement. The Cúram Express Rules language has been designed with this specific task in mind. It works on the premise that every rule is based on a question and an answer to that question. Related questions can be grouped together, and a hierarchy of question-dependencies can be established.

In determining eligibility and entitlement for a product, there can be any number of questions asked of potential product recipients in order to establish their needs. A good rules designer will create effective rules - rather than creating a flat list of many questions, the rules designer will create a smaller number of questions, with the answers feeding into other calculations.

Products can have three different types of rules:

- **Determination Rules**
Used to determine product eligibility and entitlement.
- **Decision Display Rules**
Used to display information about decisions in the application, for example, a page that displays the calculations performed to determine the amount of income assistance due to a family in need.
- **Key Decision Factor Rules**
Used to highlight key pieces of real world data, for example, the highlighting of key events in a person's life such as their date of birth, when they turn 18, or when they reach retirement age.

Rules design for decision display and key decision factor rules is approached differently to eligibility and entitlement rules. For decision display rules, the design process begins with figuring out what decision information is necessary and important to be displayed. This can be done through screen designs. Once the visual requirements are defined, it is then a matter of figuring out how to extract the necessary information to populate the screens. This information can come from the eligibility and entitlement rules results, but it can also come from unique decision display rules that have nothing to do with eligibility and entitlement decisions. For example, reasons for a sanction recommendation may be derived purely from the evidence captured for a client.

For key decision factor rules, the design process requires thinking about key pieces of data that would be useful to display in a determination result. For example, it may be useful to display the increase or decrease in the number of household members on a case, or the changes in combined household income over time.

What Information Needs to be Captured for these Rules and How?

As stated above, rules design is about asking questions in an effective manner, which means grouping related questions together. It is quite likely that the answers to many questions will come from other questions. But at some point or another, the basic data required to answer all questions emerges. Real world data required to make decisions can be extracted from existing data tables on the system, such as participant records, it can be captured as case evidence, and it can be read from rate tables. Evidence can be defined using the dynamic evidence editor; rates can be stored in the rate tables.

Rule object propagators are used to link real world data to rules. The application provides out-of-the-box support for linking evidence types and rate tables to Cúram Express Rules. For example,

linking the evidence required to make decisions regarding eligibility and entitlement is as simple as linking the name of each evidence type required to the name of the eligibility and entitlement rule set.

Capturing the Evidence That Feeds Into the Rules

Most of the data used by the rules will come from evidence. This can be captured at either the Integrated Case or Product level.

For more information about capturing evidence, see the *Evidence Guide*.

What Results Need to Be Conveyed and How?

Information that has resulted in any decisions made is of significant value to an agency, particularly as those decisions change over the lifetime of a case. Coinciding with the requirements to determine which families are eligible to receive benefits and services are the requirements to display how decisions are made, what information has been used to determine a family being eligible or ineligible, as well as the calculations that have gone into determining entitlement amounts.

Note:  The key decision factors feature is deprecated. For more information, see the *Product Overview Guide*.

The ability to see the information used to make decisions, particularly looking across the lifetime of a case, becomes invaluable in understanding how changing needs are met over time. Caseworkers and the families themselves need to know not only the benefits they are eligible to receive (or not eligible to receive) and for how long, but also the reasons why they are (or are not) eligible.

Taking a closer look at the example of providing income assistance to families in need, income assistance amounts can change over the lifetime of a case. For example, the income assistance amount may go down when one of the children in the household reaches age 18. The income assistance may go up while a family member is pregnant, and down again after the newborn baby reaches a certain age. It is important that caseworkers and families can see the 'big picture', not just after changes have occurred, but as soon as information is available. Milestone ages for family members can be displayed as key decision factors in determination results over the lifetime of a case - this can include projected ages in the future, thus providing a family with a view into what's to come. The calculations performed to determine entitlement amounts can be displayed on decision details pages, thus providing clear explanations for how amounts were derived.

Securing the Product

Security is always an important subject when it comes to managing large amounts of data, some of it quite sensitive, and for a public whose privacy is legally protected. The general topic of security falls outside the scope of this guide.

For more information, see the *Server Developer's Guide* and the *System Administration Guide*. The following sections provide a high level look at some of the security considerations to think about before building a product.

How Will Sensitive Data Be Protected?

This is the most time consuming aspect to securing a product. It involves identifying sensitive data and determining the best way to secure that data. For products, sensitive data is captured as evidence, is extracted from existing tables and used by rules (e.g. participant data) and is displayed in decisions, including new decision details pages, and as key decision factors.

Some data is secured at the page level, some is secured at the entity level, and some secured at the attribute level. For example, information about a household's income may be accessible to a wide range of users, but with the actual income amounts visible to only a small range of those users.

Security is not only about ensuring only authorized users can see sensitive data. It can also be about protecting that data from being changed by unauthorized users. For example, it may be possible for a range of users to see a person's address history, but only certain users will be authorized to maintain address information.

What Impact Do Product Changes Have on Existing Cases?

Changes to product configurations, rules, rates, and rule object propagators can impact a large number of cases. It makes sense to think about how this information is secured, particularly which users have the ability to make changes to aspects of the product that can have a large amount of impact.

Product security can be used to secure the ability to maintain product configurations: a product security identifier can be added for product maintain rights such that only users with that security identifier assigned to their user role will be able to reconfigure that product.

The accessing of evidence at the integrated case or product level is governed by product security. Evidence maintained at the product level is governed by the product security settings on that product. For evidence maintained at the integrated case level, the evidence may apply to one or more products. The security for each of these products is looked up and the most liberal product security settings apply.

Should Only Certain Users Approve Product Deliveries?

High-cost products may require an extra layer of security such that only certain users have the ability to approve these cases for activation. A product security identifier can be added to the product approve rights such that only users with that security identifier assigned to their user role will be able to approve cases for activation. In order for product approval security to work, product approvals must be set up.

For more information on configuring product security, including product approvals, see the *Integrated Case Management Guide*.

Will Evidence Be Shared across Products?

Evidence can be maintained at the integrated case level and shared across products within that integrated case. In this instance, it is important to think about the different products secured within the one integrated case.

Evidence can also be shared across agencies using the Evidence Broker. Here security issues can involve additional systems and/or organizations adding a much more complex layer to security requirements.

Security related to evidence maintenance is centralized in the Evidence Application Programming Interface (API). If the evidence being maintained is at the product level, the API ensures the evidence type is linked to that product as well as ensuring the user has maintenance rights for that product. If the evidence being maintained is at the integrated case level, the API retrieves the list of products to which the evidence applies. If the user has maintenance rights for at least one of these products, they are allowed to proceed. Evidence being shared across cases via the Evidence Broker is subject to the same security checks.

Testing the Product

The recommended approach to testing the product is to do so along the way, as testing should be an integral part of the development process. Testing options are covered throughout this guide alongside the samples on building various aspects of the product. This section provides a high-level summary of the key testing considerations for building products.

Testing Rules

The CER Rules Editor validates that the rules being written are structured correctly according to the CER definitions. This tool ensures that rules expressions are properly nested and that related expressions follow a logical path. It checks for conflicting rules that would make it impossible to reach a result.

CER itself comes with a rules coverage tool which checks the overall percentage of rules that are successfully executed by unit tests. Fundamental to the success of rules development is the creation of unit tests that automate the testing of rules. These unit tests should execute the rules, providing demonstration data that can be used to test the rules results. The rules coverage tool displays the parts of a rule set that are successfully executed by unit tests and gives an overall percentage of rules executed. Tests can be written for individual rules themselves, meaning that the building blocks of a rule set get tested in isolation. Higher level integration testing is also possible by testing group of rules, with lots of data being input and a detailed check being performed on all the possible outputs. For more information about instructions and samples on writing these unit tests and using the rules coverage tool, see the *Cúram Express Rules Reference Manual*.

Testing Results

Perhaps the most obvious test in building a new product is to check that the rules return expected eligibility and entitlement results based on the data captured. The most practical way to do this is to see if it is possible to create a product delivery for the new product and perform a manual

eligibility check. If the product and its rules are configured correctly, the output of the rules will deem the client either eligible or ineligible.

If no rules are associated to the product for a given period, the determination will include a decision of 'Eligibility could not be determined' for that period. If an error occurs, then no determination will be created and logs can be interrogated to see what the underlying problem is.

In addition to checking that the rules return expected eligibility and entitlement results, it is also important to check that the key decision factor and decision display rules return the expected visual results. The manual eligibility check can also be used to check the display output for decision details rules, but an assessment determination created during case activation must be used to check key decision factors.

Testing Rule Object Data Configurations

By default, errors in rule object data configurations (e.g. non-existent database tables or evidence types) are logged as warnings in the console output.

The treatment of these errors is configurable, by setting the environment variable `ENV_RULEOBJECTPROPAGATION_CONFIGURATION_ERROR_LEVEL` to one of:

- Error - reported as an error, processing cannot continue and is rolled back
- Warn - (default) reported as a warning, processing continues
- Ignore - not reported

Certain entity operations cannot be reliably used to identify precedent changes (for processing by the Dependency Manager). If such entity operations occur, then by default a warning is logged advising the operator to take steps to manually identify cases to that must be reassessed.

1.3 Building a Simple Product

This section provides a walkthrough of the tasks that are required to build a simple product. It goes through each of the constituent parts and explains what is required for each one. A very simple product is used throughout the section to demonstrate each stage in building a new product.

Please be advised that examples used in this section are not meant to be used in the implementation of a real product. Their sole purpose is to get you started in understanding the fundamentals of product design. One of the main purposes of using the examples in this guide is to show how the various components of the product are used in eligibility and entitlement processing. To allow rule objects to be populated by rule object converters and made available to rules processing, the naming of the rule attributes needs to align with the naming of the data attributes.

Real product design requires extensive knowledge of application development and this guide is by no means the single-source for all knowledge required to build a product. A suite of documents are available that provide detailed information of some of the topics touched in this guide. For detailed information on writing rule sets, see the *Working with Cúram Express Rules Guide*.

For detailed information on eligibility and entitlement processing, see the *Inside Eligibility and Entitlement Using Cúram Rules Guide* (this guide covers in detail the rule classes and rule

object converters and propagators used in eligibility and entitlement processing). For detailed information about designing evidence, see the *Designing Evidence Solutions Guide*.

This section uses examples to demonstrate how to build a simple product.

Simple Product Requirements

As described in the previous chapter, the first step in product design is to work out the very basic requirements that the product sets out to achieve. This section walks through the requirements for a simple product.

Purpose

The purpose of the simple product is to determine the periods during which the claimant is eligible and to figure out how much the claimant is entitled to receive in a monetary amount during these eligible periods.

Rules Required to Achieve Purpose

The very basic questions that need to be answered are:

- Who is the claimant on the case?
- What is the case start and end date (if applicable)?
- During the case lifetime, when is the claimant eligible?
- During those eligible periods, what amount is the claimant eligible for?

Information Required by these Rules

The information that is necessary to answer the above questions includes:

- Information about the claimant, e.g. if they're alive
- Information about the case, specifically the case ID and the case start and end dates
- Information that says when the claimant is eligible (and when the claimant is not eligible)
- Information that says how much the claimant is entitled to during periods of eligibility

Results to Be Conveyed

The results to be conveyed are very simply the eligible and ineligible periods for the product across the case lifetime and the entitlement amount during these eligible periods.

Essential to Product Building - Use What You Already Have

The three main tasks in product design are to design the rules that will be used to determine product eligibility and entitlement (and explain it - which is described in the next chapter), to

figure out the data that the rules require, and to map the data to the rules using rule object data configurations.

A very important part of building a new product is making the most of what you already have, rather than reinventing the wheel. The application provides out-of-the-box rule classes to get you started in designing eligibility and entitlement rules, it provides entities for capturing basic data about claimants and cases, and it provides rule object data configurations that link up rules and data types. The majority of the work required to implement the simple product can be handled by these readily available materials.

The information required to determine eligibility and entitlement for our product is quite basic: we need to know if and when the claimant is eligible and the amount the claimant is eligible for. A simple product evidence entity can be created to capture this basic data. We will then need to extend our rules to recognize this data and we will need to configure rule object data configurations to link that data to the rules.

Note: An extensive range of materials are provided, much of which falls outside the scope of the simple product discussed within this chapter. For more information about the default materials, see the *Designing Evidence Solutions Guide*.

Creating a Simple Product Evidence Entity

The information that we require to determine eligibility and entitlement is temporal in nature: the claimant can be eligible or ineligible for periods of time, and the entitlement amount during those eligible periods can change over time.

The following table describes the attributes for the new simple product evidence entity:

Table 2: Description of Simple Product Evidence Attributes

Simple Product Evidence Attribute	Description
startDate	Start date of claimant's eligibility period.
endDate	End date of claimant's eligibility period.
eligibleInd	Flag to indicate whether the client is eligible or ineligible, true being eligible and false ineligible.
amount	The entitlement amount during an eligibility period. This cannot be 0 when the eligibleInd is true. This field can also change over the course of a claimant's eligibility period. For example, the claimant may be eligible from January 1, 2011 to December 31, 2011, with differing entitlement amounts over that time period.

How to Create the Simple Product Evidence Type

The following provides a high-level overview of the steps to take in order to create the new simple product evidence type through the Admin Application:

- From Dynamic Evidence, click on the 'New' button and enter the mandatory fields in the dialog provided, namely 'Evidence Type', 'Logical Name', 'Effective From' and 'Security Group'.

- From the saved evidence type, edit the metadata for the type from the row level action menu in the version history.
- The metadata for the new evidence type is entered via the new dynamic evidence wizard.
- Different types of attribute can be entered via the palette provided. Options for each attribute, including validation, can also be specified.
- The editor allows a user to specify the user interface layout via the User Interface tab, including the number of clusters, which fields appear in those clusters etc.
- When the evidence type has been configured, it can be published thereby making it available for use.

For more detailed information on using the dynamic evidence editor, see the *Configuring Evidence Guide* guide.

Linking the Simple Product Evidence Type to Rules

When the new Simple Product Evidence Type is created and published using the dynamic evidence editor, a new code representing this type is added to the EvidenceType code table. Also created is a data rule set representing the new evidence type together with a data configuration specifying that case evidence records of this dynamic evidence type should be used to populate instances of the generated data rule class.

For more information, see [Creating the Product on page 24](#).

Using and Extending Rule Classes Included with the Application

The ProductEligibilityEntitlementRuleSet contains the fundamental rule classes required by the Cúram Eligibility and Entitlement Engine to return eligibility and entitlement results. This rule set is essential for creating the determination result for any product delivery case. The ProductEligibilityEntitlementRuleSet contains the rule classes which are the "interface" between the Eligibility and Entitlement Engine's Java code and the rules for a product.

The DefaultProductEligibilityEntitlementRuleSet contains a set of rule classes that extend the rule classes in the ProductEligibilityEntitlementRuleSet. Any classes that are temporal in nature need to extend the ActiveSuccessionSet rule class in the out-of-the-box PropagatorRuleSet.

The following is a description of the rule classes that you will use in almost any eligibility and entitlement rule set in order to return an eligible result and an entitlement amount:

Table 3: Description of Eligibility and Entitlement Rule Classes

Rule Class	Description
DefaultProduct	This rule class provides a default implementation of the AbstractProduct class found inside the ProductEligibilityEntitlementRuleSet. Rule set developers for CREOLE-based products should extend this default implementation rather than extending the ProductEligibilityEntitlement rule classes directly as these implementations provide some isolation against future changes to the rule classes in the ProductEligibilityEntitlement rule set. For more information on extending this class see Extending the DefaultProduct Rule Class on page 21

Rule Class	Description
ActiveSuccessionSet	<p>All instances of evidence populated by the ActiveSuccessionSetRuleObjectPropagator must inherit from this rule class, and the attributes which govern the start and end date of the succession set must be annotated with SuccessionStart and SuccessionEnd tags respectively. The <i>active</i> evidence rows contributing to an evidence succession set will be combined and used to populate a single rule object. For more information on extending this class see Extending the ActiveSuccessionSet Rule Class on page 21 .</p>
DefaultCase	<p>This rule class provides a default implementation of the AbstractCase class found inside the ProductEligibilityEntitlementRuleSet. As with DefaultProduct, rule set developers for CREOLE-based products should extend the default implementation rather than extending the ProductEligibilityEntitlement class directly. DefaultCase is essentially responsible for determining eligibility/entitlement within one legislative period. Extending the DefaultCase rule class provides the ability to derive the eligibility periods and entitlement amounts for the claimant over the lifetime of the case, as well as the objectives the claimant is due to receive. (see Extending the DefaultCase Rule Class on page 22).</p>
CaseParticipantRole	<p>CaseParticipantRole evidence records details of a participant associated with a case. A participant can have multiple case participant roles in a single case. A participant can have many different roles in different cases. A case may have one or more case participant roles. The Entity Rule Object Converter is configured out-of-the-box to use the data on the CaseParticipantRole database table to populate instances of the CaseParticipantRole rule class. This rule class is not used by our simple product.</p>
Person	<p>Person evidence records details of an individual about whom the organization wants to record information. As with CaseParticipantRole, the Entity Rule Object Converter is configured out-of-the-box to use the data on the Person database table to populate instances of the Person rule class. This rule class is not used by our simple product.</p>

Important: The DefaultProductEligibilityEntitlementRuleSet is a good source for understanding how eligibility and entitlement rule sets are structured and what the minimum requirements for getting a product up and running are.

Extending the DefaultProduct Rule Class

When building any product that requires an entitlement amount, you must define at least one objective type and one objective tag type. An objective is the output of a rules execution which represents, at a business level, what is being delivered to the claimant if they are deemed eligible.

For benefit products, this will typically be a form of payment, for example, Max Child Allowance. An objective tag represents the rate at which the objective will be delivered, for example, weekly rate, daily rate and so on. Objectives are used in the creation of financial schedules. The objective type indicates that the eligibility result will be a numerical amount. The objective tag is used to calculate the entitlement amount based on a frequency. During the creation of financial components, the objective and objective tags are used to create the financial schedules for eligible claimants.

In our simple product, you must extend the DefaultProduct rule class to include a SimpleObjectiveType and a DailyObjectiveTagType. The DailyObjectiveTagType requires the following attributes:

Table 4: Attributes for DailyObjectiveTagType

Attribute	Description
tagTypeID	This represents the unique identifier for the tag type.
name	The name of the tag type, 'Daily Rate' for example.
pattern	This attribute must be a valid frequency pattern; for example, daily would use the value "000100000".
valueType	This attribute is defined using the codetableentry table="RulesTagType". For our simple product, the value type code is RTT3, Money.

Extending the ActiveSuccessionSet Rule Class

The ActiveSuccessionSet Rule Class must be extended to include rules attributes that exactly match the attributes for the Simple Product Evidence.

The following attributes must be added to the class which extends ActiveSuccessionSet:

Table 5: Attributes for ActiveSuccessionSet Rule Class

Attribute	Description
startDate	This date value represents the start date of the evidence in a succession set.
endDate	This date value represents the end date of the evidence in a succession set.
eligibleInd	This Boolean attribute is defined as a timeline of Booleans and is used to derive the eligibility period for the claimant in conjunction with the start and end date (see Deriving the Eligibility and Entitlement Timelines on page 22).

Attribute	Description
amount	This number attribute is defined as a timeline of Numbers and is used to calculate the entitlement amount for the case over eligible periods. This amount can change during an eligibility period, i.e., it does not have to be the same from the start to the end date, but can vary along the timeline of the succession set record.

The ActiveSuccessionSet rule class also requires a SuccessionSetPopulation annotation which is used to indicate the lifetime of a piece of evidence. For the simple product, the start and end date values mark off the start and end dates of the succession set:

```
<Annotations>
  <SuccessionSetPopulation
    endDateAttribute="endDate"
    startDateAttribute="startDate"
  />
</Annotations
```

Extending the DefaultCase Rule Class

DefaultCase is responsible for determining eligibility/entitlement within one legislative period.

The calculation of the isEligibleTimeline attribute is covered in detail in the *Inside Eligibility and Entitlement Using Cúram Rules Guide*. For the simple product, the DefaultCase rule class must be extended to derive the eligibility timeline. The class representing an objective type will typically include the derivation of an isEntitledTimeline attribute, specifying under which conditions the claimant is entitled to the objective.

Important: The construction of timeline operations falls outside the scope of this document. For information on how to construct the timeline operations for the attributes described in this section, see the *Cúram Express Rules Reference Manual*.

Deriving the Eligibility and Entitlement Timelines

The DefaultCase rule class must be extended to include the derivation of the isEligibleTimeline attribute. This attribute is used to derive the eligible periods over the lifetime of a simple product delivery case using the startDate, endDate, and eligibleInd values for active simple product evidence.

The basic derivation of this attribute is outlined below:

- **isEligibleTimeline**

This attribute calculates the succession set periods for any simple product evidence recorded. It searches for startDate values, and endDate values, if available, of active simple product evidence. If found, it builds the list of dates that are part of the succession set.

The DefaultObjectiveType class in the DefaultProductEligibilityEntitlementRuleSet must be extended to provide a definition of the objective being delivered. It must also provide an isEntitledTimeline attribute complete with derivation. The derivation of this attribute for out simple product is outlined below:

- **isEntitledTimeline**

The claimant will be entitled to the SimpleObjectiveType if they're eligible (i.e. for periods where the isEligibleTimeline is true) and where the evidence amount is greater than zero. In fact, this processing is controlled centrally in CER. A claimant can never be entitled to benefit during a period of ineligibility. A separate amountTimeline attribute should be created to build up the list of amounts for each eligible period as determined by the isEligibleTimeline.

Configuring Rule Object Converters Included with the Application

Data configurations for rule object converters must be created to link the rules to the information required to make determinations. These configurations are used to create rule objects for the data that is required by the rules in their execution. All data configurations are in Extensible Markup Language (xml) format. The following table describes at a high level the rule object converters used by the simple product.

Note: The table below does not show all rule object converters included with the application. It only covers those used in the simple product. For detailed descriptions of all rule object converters included with the application, see the *Working with Cúram Express Rules Guide*.

Table 6: Description of Rule Object Converters Included with the Application

Rule Object Converter	Description
Entity	This populates rule objects from standard single-field-key database tables. It has been configured to create rule objects for persons and case participants.
ActiveSuccessionSet	This populates a single rule object from a succession set of evidence rows. This must be configured to create rule objects for the new simple evidence type (see Extending the ActiveSuccessionSet Rule Class on page 21).
ProductDelivery	The populates a rule object for a product delivery case for a CER-based product. The behavior of this converter is fixed - there is no configuration required.

Configuring the ActiveSuccessionSet Rule Object Converter

The ActiveSuccessionSet rule object converter retrieves timeline-aware rule objects and populates these rule objects with matching data. This rule object converter needs a data configuration in order to be able to access rule objects for the Simple Product Evidence.

The xml for this is shown below:

```
<?xml version="1.0" encoding="UTF-8"?>
<propagator type="ROPT2005">
  <configuration>
    <evidence type="ET99901">
      <ruleset name="SimpleProductEligibilityEntitlementRuleSet"/>
    </evidence>
  </configuration>
</propagator>
```

</propagator>

In the above xml, 'ET99901' is the evidence type code for the simple product evidence type (see [Linking the Simple Product Evidence Type to Rules on page 19](#)) and 'SimpleProductEligibilityEntitlementRuleSet' is the name of the eligibility and entitlement rule set.

Configuring the Product in the Administration Workspace

This section walks through the basic steps to take in configuring the simple product. It is important to note that this section is not exhaustive - it does not cover every configuration option available when setting up a new product. For a more thorough overview on product configuration, see the *Integrated Case Management Guide*.

Creating the Product

Products are now created using the dynamic product wizard. As part of product set-up it is important to specify a unique name, as well as associating integrated case categories with the new product (typically there will only be one) and defining the eligible participant types. A default eligibility and entitlement rule set is automatically created for a benefit product created by the dynamic product wizard. The default name and display name of this rule set can be edited within the product wizard. The rule set name must comply with the rule set naming standards as defined within the rule set schema. The display name can be used to record a translatable name for the rule set.

Associating Integrated Case Categories

An integrated case category can be associated with the simple product using the new product wizard or else added from the simple product home page using the actions menu. These category associations are used as filters during case creation. As part of creating a new case, an integrated case category is selected. This defines the basic integrated case information for the integrated case, including which product delivery case types can be created within the integrated case type.

It is important to associate the correct integrated case category with any new product so that new product delivery cases can be created from within the correct integrated case type. In theory, many integrated case types can be associated with a product but typically there will only be one.

Note: A standalone product delivery can be created which does not sit within an integrated case type. In this instance there is no need to associate an integrated case type with the product during creation.

Defining the Product Home Pages

There are two main home page types to be considered when configuring a new product: the integrated case home page and the product delivery case home page. As integrated cases and products can now be created dynamically without specifying home pages, both fall back on default pages which are included with the application if no page references are specified. The

respective pages are DefaultIC_home and DefaultICProduct_home. New case home pages for both the integrated case and product can be specified as part of editing the configurations.

Please note that designing the home page for a new product is outside the scope of this document. For information about UIM page design, see the *Web Client Reference Manual*.

Defining the Eligible Participant Types

A fundamental requirement in building a new product is to define who is eligible to receive that product. As part of configuring the new product, it is necessary to select all eligible participant types. For the simple product, the only eligible participant type is person.

Rules Configuration

When a new product has been created via the dynamic product wizard, important additional information can be configured via the Rule Sets tab, namely display categories, product periods and associated rules, and eligibility determination settings. A default eligibility and entitlement rule set is automatically created for a benefit product that is created via the dynamic product wizard.

A product period will automatically be created and published for the benefit product using the product structure rule class and eligibility and entitlement rule class defined within the default eligibility and entitlement rule set that is automatically created for benefit products. The processing to create and publish both the default eligibility and entitlement rule set and the product period for a benefit product is carried out in a deferred process. This deferred process is initiated at completion of the dynamic product wizard for a benefit product.

Defining Display Categories

Display categories are used to configure both the categories and the pages for case determination decision details rules. An example of a category might be 'Household Income' and the page defined may be used for displaying the output of a decision details rule set that explains the breakdown of household income per household member, and household expenses possibly.

In order for details to be displayed for a decision within a Product Delivery case, there must be at least one category defined and this can be the basic CREOLE category and display page (CREOLEDisplayRules_basicCaseDisplay) that will list the objectives for which the client is eligible. The basic page is all that is required for the simple product. This is discussed in greater detail in the next section.

Defining Product Periods and Associated Rules

A product can have one or more product periods, each of which defines a set of rule configurations for the duration of the period. A default product period is inserted for a benefit product that is created by using the dynamic product wizard. The product period requires a product structure rule class to be specified in addition to an eligibility and entitlement rule class.

Note:  The key decision factors feature is deprecated. For more information, see the *Product Overview Guide*.

The product structure rule class and eligibility and entitlement rule class for the default product period are defined within the default eligibility and entitlement rule set that is created automatically for a benefit product at completion of the dynamic product wizard.

Although the default product period is published upon creation of the benefit product created by using the dynamic product wizard, this default eligibility and entitlement rule set is left in an in-edit state following completion of the product wizard. This means that the rule set appears in the list of rule sets available for publication on the Cúram Express Rule Sets page of the Administration Workspace. This rule set is a generic rule set which is not suitable for product use prior to update. The rule set should be edited within the rule set editor to meet product requirements prior to use of the newly created product. The product structure class contains information pertaining to the product such as the types of objectives and objective tags available. The eligibility and entitlement rule class determines if a client is eligible and how much they are eligible for. The eligibility and entitlement rule set will ensure that a simple product objective is payable to the client if the client is deemed eligible.

A key decision factor rule set and decision details rule sets may be specified optionally. The key decision factor rule set is used to display the key factors related to the determination of the client's eligibility and entitlement, for example, displaying the dates on which the amount on the simple evidence entity changed. Decision details rule sets are used to display detailed information to the end user to explain how eligibility and entitlement were determined. Key decision factors and decision details are covered in [1.4 Building Decision Details Rules on page 28](#) and [1.5 \(deprecated\) Key Decision Factors on page 37](#).

Defining Eligibility Determination Settings

The following product settings have an important effect on eligibility determination of product delivery cases:

- **Allow Open Ended Cases**
This setting indicates whether or not cases require an expected end date. When set to true, determinations can have open-ended decisions. This indicator can be set to true for the simple product.
- **Decision Summary Display Strategy**
This setting indicates the entitlement display strategy. The options available are "Total daily monetary entitlement", "Total weekly monetary entitlement" or "Total monthly monetary entitlement". By setting the display strategy to "Total weekly monetary entitlement", for example, the system will automatically display entitlement results as weekly in the 'Amount' column for eligible decisions. The amounts displayed in the 'Amount' column are taken from the objective tags for the decisions. When the display strategy is set to "Total weekly monetary entitlement", the amount will be taken from the weekly objective tag. If no weekly tag is present, the daily tag amount x 7 will be displayed.
- **Determination Comparison Strategy**
This setting indicates how the system will compare data when checking for changes in circumstances. The options available are to compare all data (including decision details rule data) or to compare eligibility and entitlement data only. For the simple product, the comparison strategy is to compare all data.
- **Reassessment Strategy**
This setting indicates what kinds of cases are automatically reassessed. The options available are to reassess all cases, or to excluded closed cases for reassessment. For the simple product, the reassessment strategy is to automatically reassess all cases.

Configuring Financial and Reassessment Processing

In order to process financials, each product requires a default delivery pattern. This is to ensure financial schedules can be created during eligibility and entitlement processing. For example, the simple product has a default delivery pattern of weekly by check. By default, payments are issued on a weekly basis using the check delivery method. Delivery patterns can be set at the case level. Each nominee can have a number of delivery patterns which can be selected for financials processing.

In order to support reassessment processing, it is necessary to define how over and under payments are processed. Over and under payments can be processed as financials within the existing case, or else in separate over or under payment cases. The 'use rolled-up reassessment' setting indicates whether changes to eligibility/entitlement discovered during reassessment will result in the creation of a rolled up over/underpayment case or a granular payment correction case. For the simple product, over and under payments are processed in payment correction cases.

Associating Evidence Types with the Product or Integrated Case

Depending on where the evidence is being managed from, the new evidence type can be associated with either the integrated case or the product. In our sample product, the evidence will be managed at the integrated case level. The dynamic evidence type created for the sample product can be associated with the integrated case via the 'Evidence Types' tab from the integrated case type home page.

Testing the Product

This section provides a high-level overview of how to test the simple product rules, rule object data configurations, and determination results.

Creating Test Classes for Rules

One of the fundamentals of writing rules is to test that they do what they're expected to. When rules are written and published, they are stored in the database. A framework for testing CER rules is provided and the factory classes for writing the tests can be generated from the rule set (the rule set will need to be extracted from the database onto the file system first).

Use the following command:

```
build creole.generate.test.classes
```

This command will generate one or more test classes into the following location:

Testing Rule Set Coverage Along the Way

When rule tests are written, a rule set coverage tool can be configured and a HTML report produced to show how much coverage the tests are providing. To capture coverage data while running tests (or an online application), set the environment property `curam.creole.coverage.logfile` (in *Bootstrap.properties*) to the location of a file.

A line containing coverage information will be appended to the file whenever an expression is calculated. To clear the coverage data, just delete this file.

To create a coverage report, run the following target:

```
build creole.report.coverage -Dfile.coverage.log=(file location)
```

A simple color-coded drillable report will be written to `.../EJBServer/build/svr/creole.gen/coverage/index.html`. For detailed information about testing your rules, see the *Cúram Express Rules Reference Manual*.

Use Logging to Test Rule Object Data Configurations

Before publishing your product, you can use the `curam.trace.ruleobjectpropagation` environment variable to log the following types of rule object actions:

- Details of a database write operation that has occurred
- Details of a database write operation that is/is not of interest to a particular converter or propagator
- Value that a converter sets on a CER rule attribute

For more information on logging, see the *Designing Evidence Solutions Guide* guide.

Testing the Determination Results

The very basic level of testing required for any product is to check that the determination results are accurately portraying the expected eligibility and entitlement based on the data captured.

Such a test for our simple product would check:

- A simple integrated case and simple product delivery case can be created.
- Simple evidence can be captured and activated.
- A manual eligibility check can be used to simulate the creation of the case determination.
- The simple product delivery case can be activated.
- Upon activation, the determination is created.
- This determination accurately reflects the eligibility and entitlement based on the activated simple product evidence.

1.4 Building Decision Details Rules

This section continues on from the previous section, providing a walkthrough of the tasks that are required to build a simple product with decision details rules. Out-of-the-box, the application provides a simple decision summary page and related subscreen pages with a ruleset implementation (`SimpleDevelopmentProductDecisionDetailsRuleSet`) to retrieve the necessary data.

These samples are used throughout this section to demonstrate how to implement decision details rules to populate decision details pages.

The out-of-the-box pages can be found in the package:

```
..\EJBServer\components\core\data\initial\blob\DynamicUIM
```

The ruleset `AbstractBasicProductDecisionDetailsRuleSet.xml` can be found at:

```
..\EJBServer\components\core\CREOLE_Rule_Sets
```

This section expands upon the example for building a simple product to include decision details rules and summary pages.

Basic Steps to Follow in Designing Decision Details Rules

When deciding what needs to be displayed on decision details pages, you need to:

- Identify the categories of display rules required (for example, Summary, Financial, Residential, Income and so on.) and configure the CER-based products to contain these categories.
- For each category:
 - Sketch out the intended screenflow in line with the capabilities of dynamic UIM. For each subscreen, identify a business object ID for the data on that subscreen. Business object IDs are required as part of the key into the query mechanism and are used for displaying data on the subscreens. For example, a list of persons displayed on a subscreen can be identified by their concernRoleID.
 - Create a concrete rule class (extending `DefaultProductDecisionDetailsRuleSet.DefaultCase`) to be the "entry point" for displaying the output for that category.
 - For each subscreen in the screenflow, create a concrete rule class (extending `DefaultProductDecisionDetailsRuleSet.DefaultCaseSubscreenDisplay`), and identify the business object ID for the data.
 - Add to the rule classes the attributes whose values should be made available for display (including Timeline values, where appropriate). For each of these attributes, annotate with `Display/DisplaySubscreen` as appropriate.
 - Implement the derivations of these attributes, which may involve additional attributes or other rule classes. Where data is required from other rule classes, its attributes should also be annotated.
 - Map the business object data to the decision display rules by creating new rule object propagators or by extending existing rule object propagators.
 - Write (dynamic) UIM pages, including XPath-like syntax to query the display XML.

Defining The Display Category

When developing a product, the most important (basic) output from a rules execution that needs to be displayed to a case worker is:

1. Is the client eligible
2. If eligible, which objectives are they entitled to, and
3. For each objective they are entitled to, display the tags for those objectives

The first two can be displayed on a summary display rules screen, with the tags being displayed on a subscreen within that screen. The out-of-the-box summary display rules screen (and related subscreens) provide these basic eligibility and entitlement summary details.

Defining The Display Page

All that needs to be displayed on the summary page is an indication of whether the client is eligible or not as well as a list of objectives (and associated tags in a subscreen) they are entitled to, if they are eligible.

Identifying the Input for the Display Rules

When a display rule category and associated screen has been decided upon, the next step is to identify the inputs to that screen. These typically comes from the eligibility and entitlement rule set, as in the below example.

Extending the Decision Details Rule Set

As identified above, the decision details rule set should contain a class which extends `DefaultProductDecisionDetailsRuleSet.DefaultCase`, as shown below:

```
<Class
  extends="DefaultCase"
  extendsRuleSet="DefaultProductDecisionDetailsRuleSet"
  name="SimpleProductDecisionDetails"
>
...
...
```

For the subscreen content, the rule set should contain a class which extends `DefaultProductDecisionDetailsRuleSet.DefaultCaseSubscreenDisplay`, as shown below:

```
<Class
  extends="DefaultCaseSubscreenDisplay"
  extendsRuleSet="DefaultProductDecisionDetailsRuleSet"
  name="SimpleProductObjectiveTimelineSubscreen"
>
...
...
```

So that the rule set can retrieve the necessary information to display on the decision summary screen and associated subscreens, it should contain the following key attributes:

Table 7: Description of attributes required by display rule set

Attribute	Description
isEligibleTimeline	To determine whether the client is eligible / ineligible, the rule set should contain an attribute which will reference the isEligibleTimeline attribute on the eligibility and entitlement rule set. The attribute is named similarly here for simplicity. This attribute will live on the <code>AbstractSimpleProductDecisionDetails</code> class and will need to include a <code>Display</code> annotation.

Attribute	Description
displayObjectiveTimelines	To retrieve the list of objectives the client is entitled to, the rule set should contain an attribute which will reference the objectiveTimelines attribute on the eligibility and entitlement rule set. As there is no business data (like concernRoleID, for example) that can distinguish each individual element in the list, the businessObjectID will instead be derived from the index of each element in the list. Again, this attribute will live on the AbstractSimpleProductDecisionDetails class and will need to include a Display annotation.
displayObjectiveTimelineSubscreens	The AbstractSimpleProductDecisionDetails class will also need to contain an attribute for listing the subscreens for the objectives. This will need to contain a DisplayScreen annotated to signify that it is for subscreen display.
businessObjectID	The SimpleProductObjectiveTimelineSubscreen class needs to contain a businessObjectID attribute which will be derived from the index of the list of objectives (displayObjectiveTimelines).
displayTagTimelines	As the subscreen needs to display the list of tags for each objective, the SimpleProductObjectiveTimelineSubscreen also needs to contain an attribute which will be derived from the objective tag timelines related to each objective timeline. This attribute also needs to include a Display annotation.

Mapping Business Object Data to Decision Details Rules

The business object data required by the decision display rules must be linked to those rules using rule object propagators.

As previously described in [Using and Extending Rule Classes Included with the Application on page 19](#), The application provides out-of-the-box rule object propagators. These rule object propagators have been configured to create and populate rule objects for the data required by the simple decision display rules.

Building & Configuring Dynamic UIM Pages

In order to display information, dynamic UIM pages need to be built and configured. For retrieving the information to populate display rule pages, the following generic facades are provided:

- CaseDetermination.viewDecisionDisplayRulesCategoryXML
- CaseDetermination.viewDecisionDisplayRuleCategorySubscreenXML

The input to viewDecisionDisplayRulesCategoryXML is

- determinationID
- date (i.e. the display date of the decision being viewed)
- categoryRef

The additional input parameter to viewDecisionDisplayRuleCategorySubscreenXML is

- businessObjectID

Important: The categoryRef value that you have on the display category property file must match the categoryRef that you have specified for that display category as part of configuring the rules sets for the product (see [Rules Configuration on page 25](#)).

The data returned by the above facades is in XML, which must be parsed by XPath expressions in order to access the data to populate the fields on the page. An example of an XPath expression is show below:

```
.
.
<CLUSTER
  NUM_COLS="1 "
  SHOW_LABELS="FALSE "
  TITLE="Cluster.Title.Eligibility"
>
<CONDITION>
  <IS_FALSE
    EXTENDED_PATH="/DecisionDetails/SimpleCase/
isEligibleTimeline"
    NAME="DISPLAY"
    PROPERTY="xmlData "
  />
</CONDITION>

<FIELD>
  <CONNECT>
    <SOURCE
      NAME="TEXT"
      PROPERTY="Field.Value.ClientIsNotEligible"
    />
  </CONNECT>
</FIELD>
.
.
</CLUSTER>
.
.
```

The above example of an XPath expression shown inside a snippet of dynamic UIM would display the text associated with the property Field.Value.ClientIsNotEligible if the client was not eligible.

Note: One general point that is worth making as regards display rule design is that the complexity of the rule class interactions has a direct bearing on the resultant display data XML which is stored. A complex tree of rule classes can give rise to a deeply nested set of XML elements. This complexity should be borne in mind when trying to balance the modularity of the display rule classes against the complexity of the XPath-like syntax present in the dynamic UIM files.

Associating Decision Details Rules to the Product Period

Decision details rules are assigned to product periods under the Rule Sets configuration tab for the product. Prior to associating decision details rules to the product period, at least one display category must be defined for the product. A display category is not defined by default for a product that has been created via the dynamic product wizard.

Details on how to define a display category are outlined in [Rules Configuration on page 25](#) . To associate decision details rules to a product period within a product:

1. Select the Rule Sets tab within your product. Ensure the rule set is at In Edit status by selecting the Copy For Edit option.
2. Select the Product Periods tab within the Rule Sets.
3. Expand the details for the product period.
4. Choose the option to 'Associate Decision Details Rule'.
5. To implement the out-of-the-box decision details rules, select the Basic display category and the SimpleDevelopmentProductDecisionDetailsRuleSet.BasicCase. This display category must have been defined for the product prior to becoming available for selection when associating decision details rules.

Testing Decision Details Rules

As the layout and presentation of data on a display rules page is important, creating determinations on a product delivery, either through case activation or check eligibility, will enable the testing of the display rules page. This is the best way to ensure the functionality works as expected and the data is presented correctly on the page. Like all client pages, it may be necessary to make changes to the dynamic UIM page so that the data is presented correctly and is easy to read.

For more information about the testing of rulesets, including ruleset coverage, see the *Cúram Express Rules Reference Manual*.

Additional Samples for Creating Decision Details Pages

As the decision details summary page referred to throughout this chapter is quite basic, this section provides additional, closer to real-world, samples for creating decision details pages.

Display Data Samples

Here is sample data which is the result of querying the Income category for a particular case's determination on a given date:

```
<Display>
<IncomeCategory>
  <familyTotalIncome domain="CURAM_MONEY">250</familyTotalIncome>
  <familyMemberList>
    <Item>
      <concernRoleID>1234</concernRoleID>
      <participantFullName>James Smith</participantFullName>
      <participantTotalIncome domain="CURAM_AMOUNT">100
        </participantTotalIncome>
    </Item>
    <Item>
      <concernRoleID>5674</concernRoleID>
      <participantFullName>Jane Smith</participantFullName>
      <participantTotalIncome domain="CURAM_AMOUNT">150
        </participantTotalIncome>
    </Item>
  </familyMemberList>
</IncomeCategory>
</Display>
```

Here is sample data which is the result of querying the Income category for a particular case's determination on a given date and the business object id (1234) and subscreen type:

```
<Display>
<CaseParticipantFullIncomeClass>
  <businessObjectID>1234</businessObjectID>
  <participantFullName>James Smith</participantFullName>
  <participantIncomeList>
    <Item>
      ...
    </Item>
    <Item>
      ...
    </Item>
  </participantIncomeList>
</CaseParticipantFullIncomeClass>
</Display>
```

Note: If you set the CURAM_AMOUNT domain definition, the infrastructure automatically renders a currency symbol on the page. For more information about configuring the currency symbol, placement, and spacing, see the *Web Client Reference Manual*. However, for freeform text, you must add the currency symbol to the relevant properties files at development time. For more information about these properties files, see the *Cúram Express Rules Reference Manual*.

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

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

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

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

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

Main Category Display Class Sample

This class contains attributes to be displayed on the main screen for a category. It contains a total income amount for the whole family, as well as a list of records showing the total income for each person in the family.

```
<Class
  extends="DefaultCase"
  extendsRuleSet="DefaultProductDecisionDetailsRuleSet"
  name="IncomeCategory"
>
  <Attribute name="familyTotalIncome">
    <Annotations>
      <Display domain="CURAM_MONEY"/>
    </Annotations>
    ...
  </Attribute>

  <Attribute name="familyMemberList">
    <Annotations>
      <Display/>
    </Annotations>
    <Type>
      <javaclass name="List">
        <ruleclass name="CaseParticipantClass"/>
      </javaclass>
    </Type>
    ...
  </Attribute>

  <Attribute name="familyMemberFullIncomeList">
    <Annotations>
      <DisplaySubscreen/>
    </Annotations>
    <Type>
      <javaclass name="List">
        <ruleclass name="CaseParticipantFullIncomeClass"/>
      </javaclass>
    </Type>
    ...
  </Attribute>
</Class>
```

Sample of Standard Rule Class Used by the Main Display Class

This class is used in a list by the main display category class. It holds the name, concernRoleID and total income for a given family member.

```
<Class
  name="CaseParticipantClass"
>

  <Attribute name="concernRoleID">
    <Annotations>
      <Display/>
    </Annotations>
    ...
```

```

</Attribute>

<Attribute name="participantFullName">
  <Annotations>
    <Display/>
  </Annotations>
  ...
</Attribute>

<Attribute name="participantTotalIncome">
  <Annotations>
    <Display domain="CURAM_MONEY" />
  </Annotations>
  ...
</Attribute>
</Class>

```

Subscreen Rule Class Sample

This class represents a subscreen (or expandable list) within the main category page. It contains a list of income records for a given family member. The businessObjectID contains the concernRoleID of the person whose information is contained in a given instance object of this class.

At runtime, when a person from the list on the main screen is selected, their concernRoleID is passed to the UIM for the subscreen. This then queries the system by telling it the attributeName and businessObjectID it is looking for, in this case:

```

attributeName = familyMemberFullIncomeList
businessObjectID = concernRoleID of chosen person

```

as well as the standard information such as the determinationID, category, date etc. that was required to retrieve the XML for the main page.

```

<Class
  extends="DefaultCaseSubscreenDisplay"
  extendsRuleSet="DefaultProductDecisionDetailsRuleSet"
  name="CaseParticipantFullIncomeClass"
>

  <!-- Filled with the Persons concernRoleID -->
  <Attribute name="businessObjectID">
    <Annotations>
      <Display/>
    </Annotations>
    ...
  </Attribute>

  <Attribute name="participantFullName">
    <Annotations>
      <Display/>
    </Annotations>
    ...
  </Attribute>

  <Attribute name="participantIncomeList">
    <Annotations>

```

```

    <Display/>
  </Annotations>
  ...
</Attribute>
</Class>

```

Example of Display Rules Derived From Evidence

The data that populates a display rules page typically comes from eligibility and entitlement rulesets. However, it is possible for a display rules page to show data that is not derived from eligibility and entitlement. An example of this is sanction recommendations for household members. If a household member has been recommended for a sanction, this will be accompanied by a reason for this recommendation. Even though this information may not be used in determining eligibility and entitlement, it may be useful to show this information as part of the display rules for the case.

1.5 (deprecated) Key Decision Factors

A key decision factor is a piece of information that can help caseworkers and clients understand key pieces of data used in the eligibility and entitlement results of a case. They can be any kind of data that has a direct impact on a client's eligibility and/or entitlement. This might be derived data, for example, the total household income, or it might be a piece of evidence recorded against the case, like a new medical condition. It could even be an important event, like the birth of a new child. Presenting the information graphically in this way can help case workers to focus on the most relevant information regarding the claimant's eligibility on a particular date.

Key decision factors are optional on a product. They are useful on products where the rules are quite complex and a lot of evidence is captured. If more data is captured, it means that more data can be updated. In such circumstances, being able to see at a glance all the data that was used in coming up with the current determination, and how it changed over time, is an invaluable tool for a case worker. For instance, it allows them answer queries much more quickly for clients who may want to know why their benefit payment was lower this month than last month.

This section expands upon the example for building a simple product to include key decision factors.

(deprecated) Creating Key Decision Factor Rules

You can make key decision factor information available for a determination by creating a custom `KeyDecisionFactor` rule set and associating this with the product. Your custom rule set should extend the `DefaultProductKeyDataRuleSet.DefaultCase` class and provide a derivation for the `keyDataTimelines` attribute. This attribute is responsible for collating the key decision factor information that gets displayed on the screen.

The `keyDataTimelines` attribute returns a list of `AbstractKeyDataTimeline` rule classes. Each `AbstractKeyDataTimeline` class is considered a separate category and can choose to implement the 'timeline' attribute, the `keyEvents` attribute, or both.

These attributes allow for two distinct type of key decision factor, the key events are displayed as a date on which an important event occurred, i.e. "<event> on <date>", e.g. "Robert Smith turned 18 on 10/10/2011". The timeline attributes are displayed as a series of changes, i.e. "Changed from <old value> to <new value> on <date>", e.g., "Total household income changed from 100 to 150 on 10/10/2011".

For our simple product, the amountTimeline attribute in the eligibility and entitlement rule set could be referenced as a key decision factor 'timeline' attribute to allow a user see, at a glance, the changes to the amount over time.

(deprecated) Associating Key Decision Factor Rules to the Product Period

As in the case of decision details rules, key decision factor rules can be assigned to a product period under the Rule Sets configuration tab for a product.

(deprecated) Testing Key Decision Factor Rules

If the key decision factor rules are reusing timeline attributes from the eligibility and entitlement timeline, those rules already may have tests written for them. Using the framework for testing CER rules is covered already in this guide.

For more information, see [Creating Test Classes for Rules on page 27](#)).

Notices

Permissions for the use of these publications are granted subject to the following terms and conditions.

Applicability

These terms and conditions are in addition to any terms of use for the Merative website.

Personal use

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of Merative

Commercial use

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of Merative.

Rights

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

Merative reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by Merative, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

MERATIVE MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

Merative or its licensors may have patents or pending patent applications covering subject matter described in this document. The furnishing of this documentation does not grant you any license to these patents.

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

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

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

The licensed program described in this document and all licensed material available for it are provided by Merative under terms of the Merative Client Agreement.

COPYRIGHT LICENSE:

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

Privacy policy

The Merative privacy policy is available at <https://www.merative.com/privacy>.

Trademarks

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

IBM®, the IBM® logo, and ibm.com® are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide.

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

Oracle and Java are registered trademarks of Oracle and/or its affiliates.

The registered trademark Linux® is used pursuant to a sublicense from the Linux Foundation, the exclusive licensee of Linus Torvalds, owner of the mark on a worldwide basis.

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

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

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