IBM Cúram Social Program Management

Cúram Evidence Generator Business 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 an introduction to the Evidence Generator. This guide outlines the concepts of generating evidence adhering to the Temporal Evidence design pattern(s), gives a non-technical explanation of how the evidence generator works, and discusses what the generator can and cannot be used for.

For a technical guide to the evidence generator, see the developers guide: `Cúram Evidence Generator Cookbook`.

1.2 Audience

This guide is intended for anyone interested in learning about the Evidence Generator. It is expected that it will be read mainly by business analysts working within a social enterprise organization. It is assumed, therefore, that you are familiar with the concepts of Social Enterprise Management (SEM), and understand the organization's business requirements.

1.3 Prerequisites

To get the most from this guide, you should have a basic understanding of how the application is used by the organization to deliver benefits and services. For a high level overview see the `Cúram Integrated Case Management Guide`.

It would be helpful to understand Temporal Evidence and how it is used by the application to store information and inform decisions on case eligibility and entitlement to services. For an introduction to Temporal Evidence, please refer to the `Cúram Temporal Evidence Guide`.

1.4 Chapters in this Guide

The Evidence Generator

This chapter discusses the basic functionality of the Evidence Generator, what it requires as input, and what it produces. This chapter also discusses what can and cannot be achieved with the Evidence Generator.

Patterns

This chapter describes the characteristics of each Evidence Generator pattern to help you recognize them when you analyze the requirements for your own custom evidence types.

Conclusion

This chapter includes a summary of the main points of this guide, and a list of recommended guides for additional information.

Chapter 2

The Evidence Generator

2.1 Introduction

This chapter discusses the basic functionality of the Evidence Generator, what it requires as input, and what it produces. By understanding what it requires as input, you should have a clear idea of what files the developer has to produce to use the Evidence Generator. This chapter also discusses what can and cannot be achieved with the Evidence Generator.

2.2 Why Generate Evidence Entities?

Much of the time and effort put into custom modules goes into the design, development and maintenance of custom evidence entities and the screens necessary for capturing the evidence. With Temporal Evidence, every custom entity must implement the evidence interface, which means that there is a great deal of repetition in the code used to create and maintain evidence entities. The Evidence Generator has been designed as a tool to greatly reduce the repetitive work required to build custom evidence entities, while ensuring that all evidence entities developed meet the Temporal Evidence standards.

With the Evidence Generator, evidence entities are easier to create, saving the developer time and effort. The evidence entities are also far easier to maintain. Any changes that need to be made to how the entities work can be made through a single change to the Evidence Generator, rather than as many individual changes to all the entities. This can save a huge amount of time.

2.3 The Evidence Generator

The Evidence Generator takes in files prepared by developers, and from them produces all the necessary code and screens for a working evidence entity. The evidence type that is created is consistent in look and behavior with all Temporal Evidence.

The Evidence Generator is comprised of build scripts and Extensible Stylesheet Language Transformation (XSLT) stylesheets. The XSLT stylesheets are used to define how the information provided in the XML files is transformed into other formats to create all the necessary server and client data for an evidence entity.

2.4 What Goes In

To use the Evidence Generator, the evidence type information must be provided in the following files:

Rational® Software Architect Model

The evidence type has to be modeled in Rational Software Architect. The new evidence entity, its attributes and its operations must be added to the Rational Software Architect Model.

XML File

Additional server-side meta-data is recorded in an XML file. This additional information can include, for example, the evidence entity's relationship with a parent or child entity.

EUIM File

Additional client-side meta-data is recorded in an Evidence User Interface Meta-data (EUIM) file. The information that is pulled from this EUIM file is used by the Evidence Generator to build all the screens needed for maintaining an evidence entity.

Properties File

The properties file contains localization information and online help content for the evidence maintenance screens.

2.5 What Comes Out

When the Evidence Generator is run, it produces the following:

Server Code

All the necessary server code is generated for the following functionality:

- Creating a new evidence record
- Reading an evidence record
- Modifying an evidence record
- Deleting an evidence record

List functionality is also provided, enabling lists of the evidence records to be displayed on various pages.

Client Screens

The evidence client screens and views are generated, including tab configurations and dmx data necessary for evidence wizard pages. These screens use the Temporal Evidence User Interface, and are therefore consistent with the existing evidence screens.

Online Help Screens

The individual online help screen for each evidence maintenance page is generated.

2.6 What Is Not Generated

Some parts of an evidence type are not created by the Evidence Generator. These are:

Attribution Periods

Attribution Periods refer to the periods of time during which a piece of evidence is used in case assessment. The Evidence Generator does not generate the code used for attribution periods. Instead, developers must write module-specific code that calculates the attribution from and to dates.

Validations

An evidence validation is a check run on a piece of evidence to ensure that it meets the business requirements defined for the evidence type. It is still more efficient to hand craft the validations than to attempt to generate them. The Evidence Generator has functionality in place to enable the generated evidence entities to "call out" to the validations created by the developer.

Chapter 3

Temporal Evidence Patterns

3.1 Using Patterns

In designing evidence entities for large modules, the all entities have to fit into a relatively small number of patterns. These patterns are typically governed by how an evidence type, or entity, relates to another evidence type.

These patterns are the basis of the Evidence Generator. To use the Evidence Generator to create a new evidence type, you must analyze the relationships and behaviors of your proposed evidence type. Once the evidence type's relationships have been identified, the correct choice of pattern should be clear. An evidence entity can use more than one pattern.

This chapter describes the characteristics of each Evidence Generator pattern to help you recognize them when you analyze the requirements for your own custom evidence types.

3.2 Parent-Child Pattern

The Parent-Child pattern is the most commonly found pattern. It is used for entities that have a parent-child relationship. The characteristics of a Parent-Child relationship are:

- The Child entity must belong to a Parent, and cannot exist without a Parent.
- The Parent entity can have many Children.

3.2.1 Use It For

The Parent-Child Pattern should be used when you need to capture a logical relationship between two entities that meets the criteria of a Parent-Child relationship. For example, the Parent-Child Pattern is used to model the relationship between Income evidence and Income Usage evidence. An Income record for the money received by the client must exist before an Income Usage record for how that money was spent can be created. The money received could be spent to pay a number of bills, so the Income record can have multiple Income Usage records associated with it.

3.2.2 Don't Use It For

The Parent-Child Pattern is not suitable for evidence entities that have a very loose association, or where the evidence entity in the Child role can exist without a relationship to the Parent entity.

3.3 Pre-Association Pattern

The Pre-Association Pattern is another commonly used pattern. A pre-association relationship exists between two evidence entities that can live completely independently of each other until such time as they are associated with each other by a case worker.

3.3.1 Use It For

The Pre-Association Pattern should be used where a logical relationship exists between two records and to facilitate a caseworker in easily establishing this relationship. Before a caseworker can enter data about the main evidence type they will be prompted to select evidence as defined by this pre-association pattern and this therefore establishes a relationship between the two evidences.

For example, a client is a member of a household. The member details are captured in a Household Member evidence record. Over time, a new child is born to the household member. Again this is captured in a Household Member evidence record, however the mother-child relationship must also be captured using a Household Relationship evidence record. By specifying a pre-association pattern for household relationship evidence to household member evidence, when a caseworker creates a new household relationship record to capture the mother-child relationship, they are prompted to first select the household member before going on to enter the household relationship details. This shows that there is a logical relationship between the member and the household relationship; the member must be selected before a relationship and related member can be created.

3.3.2 Don't Use It For

Don't use the Pre-Association Pattern for evidence entities which are not logically related or entities which more naturally fall into a Parent-Child relationship. An associative relationship between an evidence entity and a non-evidence entity should not be recorded with a Pre-Association Pattern, but with a Related Pattern.

3.4 Related Pattern

The Related Pattern is designed around a relationship between a new evidence entity and a record that is not an evidence entity. The creation of the evidence entity is dependent on the other record existing, just as a child record needs a parent record.

3.4.1 Use It For

The Related Pattern should always be used when an evidence entity is being linked to a non-evidence entity. A common example of the use of this pattern is in the creation of a Paid Employment record being dependent on the existence of a (Core) Employment record. A Paid Employment record is an evidence entity, and a Employment record is part of the data captured for a client in the Participant Manager. The relationship between them is typically a foreign key relationship.

3.5 Search Case Participant Pattern

The Search Case Participant Pattern should be used if the creation or maintenance of your new evidence entity requires that you search for any case participant.

3.5.1 Use It For

The Search Case Participant Pattern should be used when it is necessary to associate a Case Participant with the evidence being created. For example, it may be necessary to search for the client's employer when creating Employment evidence as the employer's case participant role identifier may be stored on the client's Employment record.

3.6 Validate CallOut Pattern

The Validation CallOut Pattern makes it possible to add validations to generated evidence entities. The validations for a generated evidence entity are not created by the Evidence Generator. Instead, they are created by a developer in a separate file, and the Validation CallOut Pattern is used when designing the evidence entity to ensure that it can "call out" to the file that the validations are in.

When you use the Validate CallOut Pattern, the generated validate class is only generated once, so that it will not be overwritten on subsequent generations. Your developers should add this class to their own software versioning control system so that they may make modifications to it over time.

3.6.1 Use It For

The Validations CallOut Pattern must be used if custom validations need to be added to the generated evidence entity.

3.7 Page Hierarchy Pattern

An evidence entity can have many different relationships. For an evidence entity that is associated with many different types of record, it is important that these types of records are all accessible from the navigation bar on the evidence maintenance screens. The Page Hierarchy Pattern should be used when designing an evidence entity that can be associated with many different types of records.

3.7.1 Use It For

Use the Page Hierarchy pattern to generate the page hierarchy if an evidence entity can have many evidence types associated with it.

Chapter 4

Conclusion

4.1 Summary

- The Evidence Generator is a tool which creates all the necessary server code and client screens for a Temporal Evidence entity.
- Changes to multiple evidence entities can be made with a single change to the Evidence Generator.
- The Evidence Generator has several patterns which can be used alone or in combination to design your new evidence entity.
- Evidence entities whose relationships and behaviors do not match any of the patterns provided cannot be generated, but can still be created by a developer.

4.2 What to Read Next

For more detailed information on Temporal Evidence, see:

- Cúram Temporal Evidence Guide
- Cúram Temporal Evidence Developers Guide

For information on using the Evidence Generator, see:

- Cúram Evidence Generator Cookbook
- Cúram Evidence Generator Patterns Guide
- Cúram Evidence Generator Modeling Guide

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

Any references in this information to non-IBM 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.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trade-