IBM Cúram Social Program Management

IBM

Funded Program Management Developer Guide

Version 6.0.5

IBM Cúram Social Program Management

IBM

Funded Program Management Developer Guide

Version 6.0.5

Contents

Figures

Tables

vii

Chapter 1. Introduction

1.1 Purpose

The purpose of this guide is to describe the options for customizing the Cúram Funded Program ManagementCúram Funded Program Management (FPM) component and to provide instructions on how to customize the FPM component using these options. Customization can be distinguished from configuration in that customization allows developers to modify, extend, or replace source code to suit customer requirements. Configuration allows administrators to determine the information that is displayed on application pages or to alter the behavior of the application in certain predefined ways.

This guide describes the customization or extension points provided with the FPM component. For further information on how to uses these customization points, readers should consult the `Persistence Cookbook` where customization techniques are described in detail.

1.2 Audience

This guide is intended for developers responsible for customizing FPM.

1.3 Prerequisites

- `Persistence Cookbook`
- `Cúram Express Rules Reference Manual`
- `Cúram Funded Program Management Guide`

1.4 Further Reading

For more information about the classes, interfaces, business event interfaces and the standard infrastructural persistence events included in the FPM component, please consult the FPM JavaDoc.

1.5 Chapters in this Guide

Using Strategy Patterns to Customize FPM
> This chapter provides a brief description of the types of customization points that are available in FPM and how agencies can use them in a supported manner.

Using Events to Add Custom Processing to FPM
> This chapter describes the events that can be used by agencies to add functionality before and/or after a piece of functionality is executed.

FPM Rule Sets
> This chapter describes FPM Rule Sets.

Fund Waitlist
> This chapter provides details about the Fund Waitlist.

Chapter 2. Using Strategy Patterns to Customize FPM

2.1 Introduction

FPM provides service layer interfaces that are specifically designed for customization. A new custom implementation can be provided for any of the following interfaces, if a customer wishes to have a different strategy than the default implementation.

The default implementation of the following interfaces can be replaced with a new custom implementation, if required, by creating a new Google Guice module class and adding a corresponding entry in the MODULE table. For more information on Guice and how to use Guice bindings using a Module class, please refer to the Persistence Cookbook.

2.2 Determination of Fund Configuration

FPM provides the interface 'FundConfigurationDetermination' which is used for determining a fund configuration for a program fund.

A new implementation of this interface is required to change the way a fund configuration for a program fund is determined. For example, one may use a different fund configuration rather than the fund configuration that is active on the creation date of the obligation to determine the fund configuration. For example, if the agencies want to change the logic for applying the obligation to the fund configuration that is active during the service period applicable to the obligation, then those agencies can implement this interface to have their own logic to determine the fund configuration.

The default implementation has been provided by FundConfigurationDeterminationImpl and which determines the fund configuration that is active on the creation date when creating an obligation.

2.2.1 Interface Location

curam.fundpm.util.impl.FundConfigurationDetermination

2.3 Obligation Amount Calculation

FPM provides the interface which is used for calculating an obligation amount for each fund fiscal year, when the period for which the obligation is to be created crosses more than one fiscal year.

A new implementation of this interface is required to change the mechanism used to calculate an obligation amount for each fund fiscal year.

2.3.1 Default Implementation

The default implementation has been provided by FundFiscalYearObligationAmtCalculationImpl and which calculates the obligation amount for each fund fiscal year, when the obligation period overlaps more than one fiscal year.

To determine the obligation amount, the default implementation splits the obligation amount and the relevant amounts are applied to the respective fund fiscal years. The calculation of obligation amounts for all overlapping fund fiscal years is determined using the formula, $OA = (CYD / OD) * ROA$, Where: OA = The obligation amount for each fund fiscal year, CYD = The number of days in the obligation period that fall within the current fiscal year, OD = The number of days in the obligation period, ROA = The total requested obligation amount.

2.3.2 Interface Location

curam.fundpm.util.impl.FundFiscalYearObligationAmtCalculation

2.4 Fund Fiscal Year Line Item Reference Number Generation

FPM provides the interface 'FundFiscalYearLineItemRefNum' which is used to generate a reference number for a fund fiscal year line item.

A new implementation of this interface is required to change the mechanism used to generate the reference numbers.

2.4.1 Default Implementation

The default implementation has been provided by FundFiscalYearLineItemRefNumImpl. The default fund fiscal year line item reference number generation mechanism involves creation of reference number starting from 1024 and incrementing it by 1.

2.4.2 Interface Location

curam.fundpm.util.impl.FundFiscalYearLineItemRefNum

Chapter 3. Using Events to Add Custom Processing to FPM

3.1 Introduction

FPM raises several business events which allow customers to add custom logic at various points in the application. For details on how to add events listeners, please refer to the Persistence Cookbook.

3.2 Events

Each service layer API of the FPM raises a pre and post events and these events could be used to customize the service layer APIs. The service layer events can be categorized into

Standard Persistence Events (insert, modify, cancel and read)
> Please refer to the **6.7 *Writing listeners for automatic persistence events of the Persistence Cook Book*** for steps to handle these APIs.

Business Events
> Please refer to the **6.5 *Create an event listener and 6.6. Configure Guice of the Persistence Cook Book*** for steps to handle these APIs

Following is the list of business events. For additional details on any of the event classes listed here, please refer to the javadoc of the associated class.

3.2.1 Fund User Role

3.2.1.1 Interface Location

curam.fundpm.funduserrole.impl.CheckFundOwnerEvents

3.2.1.2 Event Interfaces

Table 1. Fund User Role Event Details

Name	When it is raised	Pre Event Data	Post Event Data
CheckFundOwnerEvents	This event is raised whenever we check for the fund ownership.	Fund user role accessor reference, program fund, organization object reference, type and user name.	Fund user role accessor reference, program fund, organization object reference, type, user name and fund owner indicator.

3.2.2 Fund Fiscal Year

3.2.2.1 Interface Location

curam.fundpm.impl.FundFiscalYearCanObligateEvents

3.2.2.2 Event Interfaces

Table 2. Fund Fiscal Year Event Details

Name	When it is raised	Pre Event Data	Post Event Data
FundFiscalYearCanObligateEvents	This event is raised whenever we want to check whether we can obligate before performing obligation or not.	Fund fiscal year accessor reference and amount to be obligated.	Fund fiscal year accessor reference, amount to be obligated and the obligation result indicator.

3.2.3 Fund Fiscal Year

3.2.3.1 Interface Location

curam.fundpm.impl.FundFiscalYearObligateAmountEvents

3.2.3.2 Event Interfaces

Table 3. Fund Fiscal Year Event Details

Name	When it is raised	Pre Event Data	Post Event Data
FundFiscalYearObligateAmountEvents	This event is raised whenever we want to perform an obligation.	Fund fiscal year accessor reference, amount to be obligated, obligation transaction type, funded item reference and type.	Fund fiscal year accessor reference, amount to be obligated, obligation transaction type, funded item reference and type.

3.2.4 Obligation

3.2.4.1 Interface Location

curam.obligation.impl.ObligationProcessObligationEvents

3.2.4.2 Event Interfaces

Table 4. Obligation Event Details

Name	When it is raised	Pre Event Data	Post Event Data
ObligationProcessObligationEvents	This event is raised whenever we would process an obligation request.	Obligation accessor reference, fundRelatedID, fundRelationType, requested obligation amount, obligation period, obligation related ID, type, case participant role ID and wait list expiry days.	Obligation accessor reference, fundRelatedID, fundRelationType, requested obligation amount, obligation period, obligation related ID, type, case participant role ID, wait list expiry days and Obligation result.

3.2.5 Obligation

3.2.5.1 Interface Location

curam.obligation.impl.ObligationCreateObligationEvents

3.2.5.2 Event Interfaces

Table 5. Obligation Event Details

Name	When it is raised	Pre Event Data	Post Event Data
ObligationCreateObligationEvents	This event is raised whenever an obligation is created.	Obligation accessor reference, program fund, obligation amount, obligation period, obligation related details, caseparticipant role details and wait list expiry days.	Obligation accessor reference, program fund, obligation amount, obligation period, obligation related details, caseparticipant role details, wait list expiry days and the result after obligation creation.

3.2.6 Obligation

3.2.6.1 Interface Location

curam.obligation.impl.ObligationUpdateObligationEvents

3.2.6.2 Event Interfaces

Table 6. Obligation Event Details

Name	When it is raised	Pre Event Data	Post Event Data
ObligationUpdateObligationEvents	This event is raised whenever an obligation is updated.	Obligation accessor reference, obligation amount, obligation period and obligation related details.	Obligation Accessor reference, obligation amount, obligation period, obligation related details and the result after obligation details are updated.

Chapter 4. FPM Rulesets

4.1 Introduction

This chapter gives details on the rulesets provided with the Funded Program Management.

4.2 Overview

During the process obligation, the first step is to retrieve funds for obligation. Funding rule set is used during retrieve funds for obligation to get a list of funds based on priority. The rule set could be classic rule set or creole rule set.

4.3 Rule Sets

The funding rule set is used in order to select the list of funds based on the priority which will be used for obligation.

Table 7. Funding Rule Set Details

Rule Set	Rule Set Type	Rule Set Source Location
Funding rule set	Classic Rule Set	EJBServer\components\FundPM\ rulesets\SELECTPRODFUNDS.xml
Funding rule set	CER Rule Set	EJBServer\components\FundPM\ CREOLE_Rule_Sets\ SELECTPRODFUNDS.xml

Chapter 5. Fund Waitlist

5.1 Introduction

This chapter explains the process of maintaining the Fund Waitlist as part of the Funded Program Management.

5.2 Details

When the *ObligateFund API* is called for processing an allocation request, there is a possibility that none of the eligible funds have sufficient budgets available for allocation. In such a case, the Funded Program Management system creates waitlist entries in the *WaitlistEntry* entity. First the system checks if each of the funds in the list have entries in *Waitlist* entity. If not there, it creates an entry. Then the system will create the entries in *WaitlistEntry*. Priority and Position are considered while creating the entries. The environment variable ENV_FUNDPM_WAIT_LIST_EXPIRY_DAYS can be used to configure the number of wait list expiry days for the wait list to be created during the obligation creation.

Chapter 6. FPM Batch Processes

6.1 Introduction

This chapter lists the batch processes which provide Cúram Funded Program Management functionality. The following sections list the Cúram Funded Program Management batch processes that are configurable in the system administration application.

6.2 ModifyFundFiscalYearStatus

This is the batch process that modifies the status of all the fund fiscal years to INACTIVE for the fiscal years whose end date is before the batch processing date. If the processing date is not entered by the user, the current system date will be used as the processing date. This batch should be run daily in order to make sure that the fund fiscal year status is accurate.

Batch Process Class and Method: The class and method for this batch process is `ModifyFundFiscalYearStatus.modifyFundFiscalYearStatus`.

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.

19-21, Nihonbashi-Hakozakicho, Chuo-ku

Tokyo 103-8510, 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 trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at http://www.ibm.com/legal/us/en/copytrade.shtml.

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

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

IBM ®

Printed in USA