

IBM Cúram Social Program Management  
8.0.2

*Appeals Development Guide*



**Note**

Before using this information and the product it supports, read the information in [“Notices” on page 5](#)

**Edition**

This edition applies to IBM® Cúram Social Program Management 8.0.0, 8.0.1, and 8.0.2.

Licensed Materials - Property of IBM.

© **Copyright International Business Machines Corporation 2012, 2022.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

© .

---

# Contents

- Chapter 1. Developing Appeals..... 1**
  - Enabling the appeal of case objects..... 1
  - Implementing the Appealable interface..... 1
  - Adding a code table entry..... 1
  - Binding the code table to the implementation..... 1
  - Implementing the client wizard..... 2
  - Enabling the appeal of another case type..... 2
  - Implementing the AppealableCaseType interface..... 2
  - Binding the code table to the implementation ..... 3
  - Updating the Appeals Client navigation..... 3
  
- Notices.....5**
  - Privacy Policy considerations..... 6
  - Trademarks..... 6



---

# Chapter 1. Developing Appeals

Use this information to extend the default features of IBM Cúram Appeals. By default, Appeals is configured for case types of product delivery, issue, and integrated case. Appeals can be extended to handle extra case types. Case objects can be configured to be appealed, rather than the parent case itself.

## Enabling the appeal of case objects

---

Complete the following tasks to allow objects on a case to be appealed, rather than the case itself. An appealable object can be anything on a case type that has a unique identifier.

### Implementing the Appealable interface

You must implement the Appealable interface. The data that is returned populates the Description column on the "Items Under Appeal" page.

There are two methods to be implemented in the Appealable interface.

```
LocalisableString getAppealObjectDescription  
(APPEALOBJECTTYPEEntry objectType, long objectID)
```

This method returns the globalized description for the object.

```
String getHomePageURI(APPEALOBJECTTYPEEntry objectType, long objectID)
```

This method returns the home page for the object.

### Adding a code table entry

You must add a code table entry to the ct\_AppealObjectType.ctx code table. You must set the Java identifier for the entry as it is used in binding the Java implementation.

```
<code  
  default="false"  
  java_identifier="EXAMPLEOBJECT"  
  status="ENABLED"  
  value="AOT1001"  
>  
  <locale  
    language="en"  
    sort_order="0"  
  >  
    <description>Exampleobject</description>  
    <annotation/>  
  </locale>  
</code>
```

### Binding the code table to the implementation

Bind the implementation of the AppealableObjectType interface to the AppealObjectType code in a Guice module as shown.

```
final MapBinder<APPEALOBJECTTYPEEntry, Appealable> mapbinder  
  = MapBinder  
    .newMapBinder(binder(), APPEALOBJECTTYPEEntry.class, Appealable.class);  
  
mapbinder.addBinding(APPEALOBJECTTYPEEntry.EXAMPLEOBJECT).to(  
  AppealableExampleobjectImpl.class);
```

## Implementing the client wizard

Implement the wizard framework that handles the creation of an Appeal case with a list of appealable objects. You use this framework to avoid any compile dependencies on the Appeals component.

### Procedure

1. Implement the `AppealableCaseType` interface for the parent case type.
2. Create the first wizard page, which presents a list of objects on the case to be appealed. This page must pass a delimited list of objects to the predefined second wizard screen (`Appeal_createWizard`). The format of the delimited list is:

```
ObjectID,ObjectTypeCode|
```

For example, "1001,AOT1|2001,AOT2|2002,AOT2|"

Typically, a `MULTISELECT` list is used on the client page, so a façade helper class is required to convert from the multiselect to this delimited format. A façade method is also required to return the wizard properties file.

3. Create the wizard properties file, defining the following details:

```
Number.Wizard.Pages=3

{FirstWizardPage}.Wizard.Item.Text=Select {ObjectType}
{FirstWizardPage}.Wizard.Page.Title=Step 1:
{FirstWizardPage}.Wizard.Page.Desc=Select {ObjectType}
Wizard.PageID.1={FirstWizardPage}

AppealDetermination_selectParticipants.Wizard.Item.Text
=Select Appealant/Respondent
AppealDetermination_selectParticipants.Wizard.Page.Title=Step 2:
AppealDetermination_selectParticipants.Wizard.Page.Desc
=Select Appealant/Respondent
Wizard.PageID.2=AppealWizard_SelectParticipants

AppealDetermination_createAppeal.Wizard.Item.Text
=Record Appeal Details
AppealDetermination_createAppeal.Wizard.Page.Title=Step 3:
AppealDetermination_createAppeal.Wizard.Page.Desc
=Record Appeal Details
Wizard.PageID.3=AppealWizard_createAppeal
```

Where `{FirstWizardPage}` is the name of a client page created in the previous step and `{ObjectType}` is the name of the object.

## Enabling the appeal of another case type

By default, the Appeals component is configured to work only with case types of Product Delivery, Issue, and Integrated Case. Complete the following tasks to enable the appeal of another case type by using the `Appealable Case Type` interface.

### Implementing the `AppealableCaseType` interface

Implement the `AppealableCaseType` interface for the new case type.

There are five methods to be implemented on the `AppealableCaseType` interface.

Use the following three methods to define the business logic for the case type:

```
boolean isContinueBenefitsEnabled(CaseID caseID);
```

This method returns true if Continue Benefits functionality should be enabled for this instance of the Case Type.

```
AppealableCaseTypeDetailsList listAppealableCaseDetails();
```

This method lists all of the case configurations for the abstract Case Type that can be configured for appeals.

```
boolean isCaseAppealable(CaseID caseID);
```

This method returns true if the case can be appealed in its current state. For example, if the case must be in a state of "Active", then implement the logic to check for this state in this method.

Use the following two methods to implement the wizard framework for appealing case objects:

```
String getCreateWizardProperties();
```

This method returns the name of the wizard properties file.

```
ClientURI getCreateWizardURI(CaseID caseID);
```

This method returns the initial screen in the wizard.

## Binding the code table to the implementation

Bind the implementation of the AppealableCaseType interface to the Case Type code in a Guice module as shown.

```
final MapBinder<CASETYPECODEEntry, AppealableCaseType> appealableCaseTypeBinder
= MapBinder
.newMapBinder(binder(), CASETYPECODEEntry.class,
    AppealableCaseType.class);

appealableCaseTypeBinder.addBinding(CASETYPECODEEntry.APPLICATION_CASE).to(
    ApplicationAppealableCaseType.class);
```

## Updating the Appeals Client navigation

Update the client configuration to show the Appeals pages for the new case type.

The following changes need to be made:

1. Add the following entries to the Workspace Section File:

```
<sc:tab id="AppealHearing"/>
<sc:tab id="AppealHearingCaseHome"/>
<sc:tab id="AppealHearingCaseHomeIC"/>
<sc:tab id="AppealHearingIC"/>
<sc:tab id="AppealHearingReviewHearing"/>
<sc:tab id="AppealHearingReviewHearingIC"/>
<sc:tab id="AppealHearingReviewHome"/>
<sc:tab id="AppealHearingReviewHomeIC"/>
<sc:tab id="AppealJudicialReviewHome"/>
<sc:tab id="AppealJudicialReviewHomeIC"/>
<sc:tab id="LegalActionsForHearing"/>
<sc:tab id="LegalActionsForImmediateDetentionDecision"/>
<sc:tab id="LegalActionsForPetition"/>
<sc:tab id="AppealDeskHearing"/>
<sc:tab id="AppealDeskHearingIC"/>
<sc:tab id="LegalActionOrganizationHome"/>
<sc:tab id="AppealSearch"/>
<sc:tab id="AppealHearingIssue"/>
```

2. Add a link to create an Appeal case to the Case menu file:

```
<mc:menu-item dynamic="true"
    id="CaseAppeal"
    page-id="{pageID}"
    title="MenuItem.Title.CaseAppeal"
    tooltip="MenuItem.Tooltip.CaseAppeal"
    open-as="modal"
```

```
/>
```

Where {pageID} is Appeal\_resolveCaseAppealWizard or, if appealing an object, the name of the first wizard screen.

## Notices

---

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

#### COPYRIGHT LICENSE:

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

## Privacy Policy considerations

---

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

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

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

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

## Trademarks

---

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

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

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

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

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

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



Part Number:

(1P) P/N: