



IBM Cúram Social Program Management

Cúram Deployment Guide for WebLogic Server

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.

Licensed Materials - Property of IBM

Copyright IBM Corporation 2012. All rights reserved.

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

© Copyright 2008-2011 Cúram Software Limited

Table of Contents

Chapter 1 Introduction	1
1.1 Deployment Guide	1
Chapter 2 Building .ear Files	3
2.1 Introduction	3
2.2 The Enterprise Application	3
2.2.1 Building the Application .ear File	3
2.2.2 Under the Hood	4
2.2.3 Contents of Application .ear File	4
2.3 The Online Help Application	6
2.3.1 Building the CuramHelp.ear File.	6
2.3.2 Contents of the CuramHelp.ear File	6
2.4 The Web Services Application	7
2.4.1 Building the Web Services .ear File	7
2.4.2 Under the Hood	7
2.4.3 Contents of Web Services .ear File	8
2.4.4 Web Service WSDL	8
2.5 Multiple EAR files	9
Chapter 3 Application Server Configuration	12
3.1 Introduction	12
3.2 WebLogic Server Configuration	12
3.3 Security Configuration	13
3.4 Time Zone Configuration	14
3.5 Starting and Stopping WebLogic Servers	14
3.5.1 Start a WebLogic Server	14
3.5.2 Stop a WebLogic Server	15
3.5.3 Restart a WebLogic Server	15
3.5.4 Server Logging	16
Chapter 4 Deployment	18
4.1 Introduction	18
4.2 Deployment	18
4.2.1 Deploy an Application	18
4.2.2 Change SYSTEM Username	19
4.2.3 Undeploy an Application	19
4.3 Pre-compiling JSPs	20

4.4 Testing Deployment	20
Appendix A Manual WebLogic Server Configuration	23
A.1 Introduction	23
A.2 Creating a WebLogic Server Domain	23
A.2.1 Domain Configuration	23
A.3 Java Virtual Machine Configuration	24
A.4 Set up Database Driver	25
A.5 Starting the WebLogic Administration Service	26
A.6 Database Configuration	26
A.7 Setup Security	28
A.8 Setup JMS	30
A.9 Queue Setup	31
A.10 Queue Connection Factory Setup	32
A.11 Topic Setup	34
A.12 Topic Connection Factory Setup	35
A.13 Manual Deployment	35
Notices	38

Chapter 1

Introduction

1.1 Deployment Guide

This guide discusses the steps necessary to build an *IBM® Cúram Social Program Management* application for deployment on *Oracle® WebLogic Server*. The guide also details the support provided for configuring and deploying on *WebLogic Server* and, where necessary, the manual steps required.

It is a prerequisite that the reader is knowledgeable about the *IBM Cúram Social Program Management* application development environment and knows how to develop and build a server application and web client. The guide also presumes that *WebLogic Server* has been previously installed. For details on the installation of *WebLogic Server*, consult the *Cúram Third Party Tools Installation Guide*¹.

Notes

¹Refer to the installation guide that is relevant to the platform, i.e., *Microsoft® Windows* or *UNIX*.

Chapter 2

Building .ear Files

2.1 Introduction

The main step before deployment of an *IBM Cúram Social Program Management* application is to package it into `.ear` (*Enterprise ARchive*) files. The application (*web client, server*) and *web services* application are all packaged into separate `.ear` files, and the *Server Development Environment for Java (SDEJ)* provides *Apache Ant* targets that perform this task.

Before the targets in the following section are executed, the `WLS_HOME` environment variable must be set, along with the previously set-up environment variables from the *Cúram Third Party Tools Installation Guide*.

The `WLS_HOME` variable should point to the `server` directory of the *WebLogic Server* installation, for example: `d:\weblogic\server` or `/opt/weblogic/server`.

2.2 The Enterprise Application

The following sections describe the process for building the application `.ear` (*Enterprise ARchive*) file. They also provide information on what happens "under the hood" and on the contents of the EAR file.

2.2.1 Building the Application .ear File

The following target should be executed from the root directory of the project to create the `.ear` file for *WebLogic Server*:

build weblogicEAR

This target will create a ready to install `.ear` file, `<SERVER_MODEL_NAME>.ear`, located in `<SERVER_DIR>/build/ear/WLS1`.

This target will also create a ready to install `.ear` file, `CuramHelp.ear`, located in `<SERVER_DIR>/build/ear/WLS`, provided the online help application has been created and built. For details on how to build the online help application, please refer to the *Cúram Online Help System Development and Deployment Guide*. For details on the contents of the `CuramHelp.ear` file, please refer to Section 2.3.2, *Contents of the CuramHelp.ear File*

Before executing this target, a fully built *IBM Cúram Social Program Management* application must be available. For details on how to build an *IBM Cúram Social Program Management* application, please refer to the *Cúram Server Developer's Guide*.



Note

The EAR file cannot be built for *H2* database.²

2.2.2 Under the Hood

The **weblogicEAR** target takes a number of previously generated *Java*® files and deployment descriptors and packages them up into an EAR file.

The *Java* files and deployment descriptors are generated during the build process based on the existence of Business Process Object (BPO) classes, i.e., the methods of *Facade* classes or `<<WebService>>` classes and can be called by remote clients.

By default all remote calls to the server are handled by the session bean `curam.util.invoke.EJBMethod`, rather than a session bean per publicly available interface. This bean provides support for application features such as authorization, auditing and tracing. If required it is also possible to generate a *Facade* interface³.

2.2.3 Contents of Application .ear File

The `.ear` file that is produced has the following structure and contents:

- **META-INF Directory.** The `META-INF` Directory includes the following:
 - `application.xml`
This file is automatically generated and lists the mapping of EJB modules to JAR files that are contained in the application.
 - `MANIFEST.MF`
This file details the contents of the `.ear` file.
- **Core .jar Files.** The core `.jar` files include⁴:
 - `appinf.jar`
 - `appinf_internal.jar`

- `coreinf.jar`
- `rules.jar`
- `jde_commons.jar`
- `log4j.jar`
- `commons-pool.jar`
- `commons-codec.jar`
- `commons-discovery.jar`
- `jdom.jar`
- `axis.jar`
- `castor.jar`
- `jaxrpc.jar`
- `saa.jar`
- `java_cup.zip`
- `InfrastructureModule.jar`
- `InvalidationModule.jar`
- `DBtoJMS.war`
- `ClientModule.war`
- **Facade .jar Files.** These are only present if facade generation has been enabled. All facades defined in the application are packaged into one .jar file, `FacadeModule.jar`. This .jar file contains the bean implementation classes for the *EJB* modules that represent the facades. The .jar file contains the following files in the `META-INF` directory:
 - `ejb-jar.xml`

This file is automatically generated and contains the definition of every *EJB* module contained in the .jar file. All the publicly available methods are listed and the details of the resources available to the *EJB* modules.
 - `weblogic-ejb-jar.xml`

Generated *WebLogic Server* specific deployment descriptor.
 - `Manifest.mf`

The manifest file, detailing the classpath for the *EJB*.
- **Other .jar Files.** The other .jar files contain the generated and hand crafted code from the application. These include `struct.jar`, `ap-`

plication.jar, workflow.jar, implementation.jar and properties.jar. The properties.jar file contains the Bootstrap.properties file. This is the file containing the machine specific configuration properties for initially getting a connection to the database.

2.3 The Online Help Application

The *IBM Cúram Social Program Management* online help application is built in a separate CuramHelp.ear file. The CuramHelp.ear file automatically gets generated when the application EAR file is created. Please refer to Section 2.2.1, *Building the Application .ear File* for further details on building the Application EAR file. The ability to build the online help application EAR file separately is also available.

2.3.1 Building the CuramHelp.ear File.

The following target should be executed from the root directory of the project to create the CuramHelp.ear file for *WebLogic Server*:

build weblogicHelpEAR

This target will create a ready to install CuramHelp.ear file, located in the <SERVER_DIR>/build/ear/WLS directory, provided the online help application has been created and built. For details on how to build the online help application, please refer to the *Cúram Online Help System Development and Deployment Guide*.

2.3.2 Contents of the CuramHelp.ear File

The CuramHelp.ear file that is produced has the following structure and contents:

- **META-INF Directory.** The META-INF Directory includes the following:
 - application.xml
This file is automatically generated and lists the mapping of EJB modules to JAR files that are contained in the application.
 - MANIFEST.MF
This file details the contents of the .ear file.
- **help.war files.** A help.war is a web application containing application online help screens. A help.war will be created for each locale supported. Please refer to the *Cúram Online Help System Development and Deployment Guide* for further details on online help.

2.4 The Web Services Application

Support is available for the automatic generation of *WSDL*⁵ defined *Web Services*. Application developers can thus combine the power of the *IBM Cúram Social Program Management* model with the accessibility of web services to produce truly reusable software components.

2.4.1 Building the Web Services .ear File

The following target should be executed from the root directory of the project to create the .ear file for web services:

```
build weblogicWebservices -Dprp.webipaddress="address" -Dprp.contextproviderurl="url" -Dprp.contextfactoryname="factory"
```

Where:

- “address” is the IP address on which the server hosting the web services is listening. The default is `http://localhost:7001`;
- “url” is the *URL* of the *JNDI* context provider. This is the address of the server that hosts the *IBM Cúram Social Program Management* components being made accessible through web services. The default⁶ is `t3://localhost:7001`;
- “factory” is the *JNDI* context factory name. The default for this is `weblogic.jndi.WLInitialContextFactory` and should rarely need to be changed.

This target will create a ready to install .ear file, `<SERVER_MODEL_NAME>WebServices.ear` located in `<SERVER_DIR>/build/ear/WLS`.



Note

Before executing this target, a fully built *IBM Cúram Social Program Management* application, ready for deployment, must exist.

2.4.2 Under the Hood

The **weblogicWebServices** target takes a number of previously generated *Java* files and deployment descriptors and packages them up into an .ear file.

The *Java* files and deployment descriptors are generated during the build process (see the *Cúram Server Developer's Guide*) based on the web service stereotypes that have been defined in the model. BPO classes should be mapped to server components with a stereotype of `<<webservice>>` for this generation to occur⁷. Any server component with a stereotype of `<<webservice>>` will be treated as if it also had a stereotype of `<<ejb>>`. This is because web service interfaces are wrappers on publicly

available BPOs.

2.4.3 Contents of Web Services .ear File

The web services .ear file that is produced has the following structure and contents:

- **META-INF Directory**
 - `application.xml`
This file details the core module for the web services application, which is the `webservices.war` file.
 - `MANIFEST.MF`
The manifest file which details the contents of the .ear file.
- **Web Service .war File.** This file contains support .jar files in the `WEB-INF/lib` directory, including:
 - `coreinf.jar`
This .jarfile contains the conversion methods which are used to support the serialization of the complex types used in the interface.
 - `axis.jar`
This .jar file contains the *Apache Axis* web services engine.
 - `appwebservices.jar`
This .jar file contains the wrapper classes which enable the *Axis* web services to connect to *IBM Cúram Social Program Management* session bean(s) and the classes for the complex types which are used in the interface to the web services.
 - `server-config.wsdd`
The .wsdd file is located in the `WEB-INF` directory and contains the web service engine configuration which maps *IBM Cúram Social Program Management* BPOs to web services.

2.4.4 Web Service WSDL

A *IBM Cúram Social Program Management Axis* web service exposes its own *WSDL* once it is deployed.

For instance, if there is a service at the *URL*:

```
ht-
tp://localhost:7001/CuramWS/services/MyTestService
```

the *WSDL* description will be at the *URL*:

```
ht-
```

tp://localhost:7001/CuramWS/services/MyTestService?
wsdl

The *URL*

http://localhost:7001/CuramWS/services

will return a web page that lists all *Axis* web services deployed and a link to their *WSDL* files.

The general *URL* format of the locations above is

ht-

tp://<web-server>:<port-number>/<ServerModelName>WS
/services/<BPO-name>.

2.5 Multiple EAR files

Building an application EAR also takes an optional file to allow for splitting the client components into different WAR and EAR files and also to allow for some more control of some of the EAR configuration and included modules. This file is named `deployment_packaging.xml` and should be placed in your `SERVER_DIR/project/config` directory.

The format of the `deployment_packaging.xml` file is as follows:

```
<deployment-config>
  <ear name="Curam"
    requireServer="true">
    <components>custom,sample,SamplePublicAccess,core</components>
    <context-root>/Curam</context-root>
  </ear>
  <ear name="CuramExternal">
    <components>SamplePublicAccessExternal</components>
    <context-root>/CuramExternal</context-root>
    <custom-web-xml>${client.dir}/custom_web_xml</custom-web-xml>
  </ear>
</deployment-config>
```

Example 2.1 deployment_packaging.xml Sample

Each file can have multiple `<ear>` elements and results in an EAR file being produced in the `SERVER_DIR/build/ear/WLS` directory. The options for each element are:

- `name`

This option controls the name of the EAR created from the process.

- `requireServer`

This optional attribute controls whether the server module is included in the EAR file. Valid entries are `true` or `false`. The default value is `false`. If deploying multiple EAR files to one application server, this attribute must be set to `true` for only one EAR file as only one *IBM Cúram Social Program Management* server module should be deployed per cluster. If `requireServer` is set to `true` for multiple EAR files,

then the other EAR files must be deployed in another cluster to avoid conflicts.

- `components`

This option controls which of the client components get placed into the EAR file. It also controls the component order for the rebuild of the client that will need to take place. Usually the core directory doesn't form part of the component order but on this occasion it is important to add this to qualify whether it should be included in a particular WAR file. Entries here should follow the typical order of components defined in the *Cúram Server Developer's Guide* and should be comma separated.

- `context-root`

This option forms the Context Root of the WAR module in the `application.xml` deployment descriptor. Entries here should begin with a forward-slash.

- `custom-web-xml`

This optional element controls whether a custom `web.xml` file should overwrite the standard version in the WAR file. Entries here should be an *Apache Ant* path to the directory containing the `web.xml` file.

It is possible to use references to environment variables as part of this path. For example, `${client.dir}` can be used to point to the web client directory and `${SERVER_DIR}` can be used to point to the server directory.

For each web client (i.e., WAR file) a separate web client component is required to contain its customizations. In the case of multiple web clients, your `CLIENT_COMPONENT_ORDER` environment variable will include all your custom components; but, separate `<ear>` elements will be required, one for each custom web component (and other components as needed).

As with the standard target, a fully built application must be available. For details on how to build an application, please refer to the *Cúram Server Developer's Guide*.

Notes

¹SERVER_MODEL_NAME and SERVER_DIR are environment variables which specify the name of the model in the project and the root directory of the project respectively.

²For more information on *H2* database consult the *Cúram Third-Party Tools Installation Guide for Windows*.

³The optional build parameter `-Denablefacade=true` turns on the generation of facade code.

⁴The version numbers are not listed for the `.jar` files detailed.

⁵*WSDL* means Web Service Definition Language.

⁶The default value for the *JNDI* context provider uses a *WebLogic Server*-specific protocol, *t3*, for accessing the *JNDI* tree.

⁷Consult the *Cúram Server Modelling Guide* for details on assigning BPOs to server components.

Chapter 3

Application Server Configuration

3.1 Introduction

This chapter presumes that *WebLogic Server* has already been installed. Consult the *Cúram Third Party Tools Installation Guide*¹ for details on the installation.

The configuration of *WebLogic Server* is the same on all platforms and the *Cúram Server Development Environment for Java* (SDEJ) provides a number of *Ant* targets to aid the configuration and management of the installation. Appendix A, *Manual WebLogic Server Configuration* details the manual steps performed by the configuration scripts.

The configuration target provided by the *SDEJ* is a simple default configuration and may not be suitable for a production environment.



Note

The **configure** target will overwrite any existing *WebLogic Server* domain that it is targeted to configure.

3.2 WebLogic Server Configuration

The configuration of *WebLogic Server* involves setting up a data source, a domain and a server and configuring the *JMS* and security settings. All these tasks can be performed by executing the **configure** target provided by the *Cúram SDEJ*.

The command **build configure** should be executed from the `<SERVER_DIR>` directory to invoke automatic configuration. This target requires that the files `AppServer.properties` and `Bootstrap.properties` exist in the `<SERVER_DIR>/project/properties`² directory. See the *Cúram Server Developer's Guide* for more information on the setup of a Boot-

strap.properties. Example 3.1, *AppServer.properties.sample* details the required contents of the `AppServer.properties` file.

```
## APPLICATION SERVER PROPERTIES

## IT IS VERY IMPORTANT TO USE '/' FOR DIRECTORY PATHS. ##

# Property to indicate WebLogic Server is installed.
as.vendor=BEA

# The username and password for admin server.
security.username=<e.g. weblogic>
security.password=<e.g. Weblogic1>

# The name of the WebLogic Server Domain Name.
node.name=MYNODE

# The name of the server on which the application will be hosted.
curam.server.name=CuramServer
curam.server.port=7001

#####
## THE FOLLOWING PROPERTIES ARE FOR WEBLOGIC ONLY ##
#####

# Property to set JVM initial and maximum heap size
# when starting and stopping WebLogic Server.
curam.server.jvm.heap.size=1024
```

Example 3.1 AppServer.properties.sample



Note

1. The `security.username` used here in the `AppServer.properties` file should not be the same as any user that will exist as an application user.

With the implementation of strong password enforcement in *WebLogic Server 11g Release 1*, the *Weblogic* application server password must be at least eight alphanumeric characters in length with at least one number or a special character.

2. The **configure** target cannot be run while *H2* database is in use.

3

3.3 Security Configuration

Authentication providers are the default authentication mechanism in *WebLogic Server* and application security is implemented using a custom authentication provider. The *Default Configuration for Oracle WebLogic Server* section in the *Cúram Security Handbook* guide should be consulted for further details on this.

The application and *WebLogic Server* support the use of alternative authentication mechanisms, such as an LDAP directory server or a single sign-on solution. *WebLogic Server* provides authentication providers that can be configured to work with LDAP directory servers and for single sign-on

solutions the third party vendor often produces a custom authentication provider to work with *WebLogic Server*. Where an alternative provider is to be used for authentication the Cúram authentication provider should only verify that users are valid for authorization purposes.

To configure the Cúram provider for identity only authentication the `curam.security.check.identity.only` property should be set to `true` in the `AppServer.properties` file before the **configure** target is run. The additional authentication providers should be configured manually after the `configure` target has been run.

The *Identity Only Authentication* section in the *Cúram Security Handbook* guide should be referenced for further details on this.

An optional property is available that will enable logging for the Cúram authentication provider. This property `curam.security.login.trace`, when set to `true`, results in tracing information being added to the *WebLogic Server* log file during the authentication process. This property should be set in the `AppServer.properties` file before the `configure` target is run.

3.4 Time Zone Configuration

If multiple server machines are used, they all must have their clocks in sync and be in the same time zone in order that the "natural" ordering of date/times on the database accurately reflects the order that the events occurred in the real world. For example if on the database record *A* has a creation date/time field earlier than that on record *B*, then we can say for sure that *A* was created before *B*, no matter which server created either record.

The time zone of the server(s) must never change during the lifetime of the application. The reason for this that the time zone assumed when storing dates in the database is the current server's time zone; therefore if the server's time zone changes then all dates entered prior to the time zone change will be out by the number of hours equal to the difference between the old and new time zones.

3.5 Starting and Stopping WebLogic Servers

For *Windows* platforms, the Cúram *SDEJ* provides *Ant* targets to aid in the starting and stopping of the *WebLogic* server. These targets should be executed from the `<SERVER_DIR>` directory and, as for the **configure** target, they require the `AppServer.properties` file to be set up correctly (see Example 3.1, *AppServer.properties.sample*). They also require a number of extra parameters to be specified. These are detailed below.

3.5.1 Start a WebLogic Server

The *Ant* target for starting a *WebLogic* server is:

build startserver

and requires the following option:

- `-Dserver.name`

The name of the server to be started.

**Important**

Before starting the application server for the first time you must have run the **database** target followed by the **pre-prepare.application.data** target. Failing to run this sequence will likely result in transaction timeouts during first login and a failure to initialize and access the application. Whenever the **database** target is rerun (e.g. in a development environment) the **pre-prepare.application.data** target must also be rerun.

```
build startserver -Dserver.name=CuramServer
```

Example 3.2 Example of Usage

3.5.2 Stop a WebLogic Server

The *Ant* target for stopping a *WebLogic* server is:

build stopserver

and requires the following option:

- `-Dserver.name`

The name of the server to be stopped.

```
build stopserver -Dserver.name=CuramServer
```

Example 3.3 Example of Usage

3.5.3 Restart a WebLogic Server

The *Ant* target for restarting a *WebLogic* server is:

build restartserver

and requires the following option:

- `-Dserver.name`

The name of the server to be restarted.

```
build restartserver -Dserver.name=CuramServer
```

Example 3.4 Example of Usage



Note

If the server is not already started when attempting to restart it, the stop portion of the *Ant* target will not cause the target to fail.

3.5.4 Server Logging

The servers are started and stopped as *Windows Services* and can be found in the Service Listing in the format `<domain_name>_<server_name>`. The default server logging from this service gets placed in the file `<bea.home>/user_projects/domains/<node.name>/servers/<server.name>/logs/<server.name>_Redirect.log`. This log gets cleared upon server start-up.

Notes

¹Refer to the installation guide that is relevant to the platform, i.e *Windows* or *UNIX*.

²It is possible to overwrite this default location for the properties file by specifying `-Dprop.file.location` when executing the **configure** target.

³For more information on *H2* database consult the *Cúram Third-Party Tools Installation Guide for Windows*.

Chapter 4

Deployment

4.1 Introduction

The final step after packaging the *IBM Cúram Social Program Management* application and web services `.ear` files is to deploy them on the application server.

The default installation for the application and web services `.ear` is to deploy them on the server in the same installation of *WebLogic Server*.

4.2 Deployment

Before deploying an application, restart (or start) the *WebLogic* server as detailed in Section 3.5, *Starting and Stopping WebLogic Servers*. When deploying the *IBM Cúram Social Program Management* application ensure that the database is configured correctly¹.

The *SDEJ* provides *Ant* targets for deploying and undeploying applications on a *WebLogic* server. As with the **startserver/stopserver** targets, the **installapp/uninstallapp** targets require the `AppServer.properties` file to be configured correctly (see Section 3.2, *WebLogic Server Configuration*). The targets also require a number of options detailed in the following sections.

4.2.1 Deploy an Application

The *Ant* target to deploy or install an application (in the form of an `.ear` file) is:

build installapp

and requires the following options:

- `-Dserver.name`

The name of the server to install the application on.

- `-Dear.file`

The fully qualified name of the `.ear` file to install.

- `-Dapplication.name`

The name to identify the application when it is installed.

```
build installapp -Dserver.name=CuramServer
                 -Dear.file=d:/ear/WLS/Curam.ear
                 -Dapplication.name=Curam
```

Example 4.1 Example of Usage



Note

The EAR file containing the server module must be deployed before installing any other (client-only) EAR files.

4.2.2 Change SYSTEM Username

It is strongly recommended that you change this username after deploying the application using the *WebLogic Server* administration console. The `Run As User` property should be changed from `SYSTEM` to the user of choice. The password of this user does not matter, since no authentication and only identity assertion is performed by *WebLogic Server* for JMS invocations.

The `Run As Principal Name` property can be found on the Configuration Tab for each of the MDB (Message Driven Bean) EJB modules deployed, and should match this value. This also requires updating `Run As User` property of *myrealmCuramAuthenticator* (Curam Authentication Provider) to new user of choice under `Security Realms` section in the *WebLogic Server* administration console. Consult the *WebLogic Server* documentation for more information on the usage of the administration console.

Note, if the username is changed, the new username must exist in the `Users` database table and this user must have a role of `'SUPERROLE'`.

The `SYSTEM` user is the user under which JMS messages are executed.

4.2.3 Undeploy an Application

The *Ant* target to undeploy or uninstall an application is:

build uninstallapp

and requires the following options:

- `-Dserver.name`

The name of the server the application is installed on.

- `-Dapplication.name`

The name of the application to uninstall (as configured during install).

```
build uninstallapp -Dserver.name=CuramServer
-Dapplication.name=Curam
```

Example 4.2 Example of Usage

4.3 Pre-compiling JSPs

An additional target, **precompilejsp**, is available during deployment. It allows for the *JSPs* of a client `.ear` file to be pre-compiled *before* installing the `.ear` file. This speeds up the display of a particular page in the web browser the first time it is accessed.

The options for the **precompilejsp** target are:

- `-Dear.file`

The fully qualified name of the `.ear` file to be pre-compiled.

```
build precompilejsp -Dear.file=d:/Curam.ear
```

Example 4.3 Example of Usage

This target will overwrite the `<Curam.ear>` with a copy that contains the pre-compiled *JSPs* and can then be deployed as described in Section 4.2, *Deployment*.

4.4 Testing Deployment

When the application is installed² on a configured *WebLogic Server* installation the application should be started and tested.

To do this, ensure the relevant server is started³ and open the following page in a web browser:

```
https://<some.machine.com>:<port>/<context-root>
```

where,

`<some.machine.com>` identifies the the host name or IP address where your *WebLogic Server* system is running, `<port>` identifies the SSL port of the server the application is deployed on⁴ and `<context-root>` identifies the Context Root of the WAR module (see Section 2.5, *Multiple EAR files*, for details).

Before the page can be opened, the browser will be directed to the login page. Log-in with a valid application username and password and the browser will be redirected to the requested page.



Note

The usage of EAR file name `Curam.ear` for option `-Dear.file`

and usage of application server name `Curam` for option `-Dapplication.name` in the examples of this chapter are for illustrative purposes. Based on your customized application and deployment strategy these values may change.

Notes

¹If the database does not contain the necessary information deployment may fail for security/validation reasons.

²The installation of a web services application may also be required.

³There is no need to restart the server after an application is deployed.

⁴The SSL port is one number up from the `curam.server.port` specified in the `AppServer.properties` file.

Appendix A

Manual WebLogic Server Configuration

A.1 Introduction

If required, the *WebLogic Server* installation can be configured manually. However, this is not recommended. For informational purposes only, this appendix details the manual steps required to configure and deploy on *WebLogic Server*.¹

This appendix presumes that *WebLogic Server* has already been installed. Consult the *Cúram Third Party Tools Installation Guide*² for details on installation.

A.2 Creating a WebLogic Server Domain

The *Domain Configuration Wizard* is a tool to aid in the configuration of a *WebLogic Server* domain.

On *Windows* Platforms the domain configuration wizard can be invoked from:

Programs+Oracle WebLogic+WebLogic Server 11gR1+Tools+Configuration Wizard

On *UNIX* Platforms the domain configuration wizard can be invoked by running:

```
<wls.home>/weblogic/common/bin/config.sh
```

where <wls.home> is the *WebLogic* installation directory (e.g. /opt/wls).

A.2.1 Domain Configuration

When the domain configuration wizard opens, follow the steps below, *exactly* as outlined, to configure the domain:

1. Select *Create a new WebLogic Domain* and click *Next*;
2. Select *Generate a domain configured automatically to support the following products*. Ensure the *WebLogic Server* checkbox is selected and click *Next*;
3. Set the *Domain name*.
The *Domain name* (and all server names) can be called anything so long as the `AppServer.properties` file is updated correctly. Leave *Domain location* as default and click *Next*;
4. Configure an Administrative Username and Password with a username of “weblogic” and password of “weblogic1”. Note that the password must be at least eight alphanumeric characters with at least one number or a special character. Confirm the password and click *Next*;
5. Choose the *Startup mode* and *Java SDK* as necessary and click *Next*;
6. Choose nothing in the *Select Optional Configuration* screen and click *Next*;
7. Review the Configuration settings and click *Create* to construct the domain.
8. The *Configuration Wizard* is now complete. Click *Done* to end the *Configuration Wizard* and exit the installer.

The *IBM Cúram Social Program Management* application requires a `MaxPermSize` of at least 256m to run. This must be configured as a parameter to the newly created domain. Edit the file relevant to your platform (located at `<WebLogic Install directory>/<domain_repository>/<DomainName>/bin/`), either:

1. `setDomainEnv.cmd` - *Windows*
2. `setDomainEnv.sh` - *UNIX*

and where it passes `MaxPermSize`, set this to 256m.



Note

The setting of 256m is for illustrative purposes. The optimum value should be determined by monitoring the memory utilized for your server.

A.3 Java Virtual Machine Configuration

The *IBM Cúram Social Program Management* application requires additional parameters to be passed to the JVM running the Application Server. Set the following system environment variables:

- `USER_MEM_ARGS` : This should be set to a value relevant to your application e.g. `-Xmx712m`. This environment variable is set by the

app_runtimewls.xml script using these properties, which can be defaulted as follows or set in the AppServer.properties properties file:

Property Name	Default Value	Description
curam.server.jvm.heap.size	1024	Specifies the JVM heap initial and maximum sizes.
curam.server.jvm.permgen.size	-XX:PermSize=128m -XX:MaxPermSize=256m	Specifies the initial and maximum PermGen sizes.

Table A.1 Memory Arguments

If you override the USER_MEM_ARGS environment variable directly you must also provide values for the JVM PermGen space as shown in Table A.1, *Memory Arguments*.

- WLS_REDIRECT_LOG : Output file e.g. <WebLogic Server Install directory>/<domain_repository>/<DomainName>/logs/<server.name>_Redirect.log
- JAVA_OPTIONS : This environment variable is used for two purposes.
 1. This environment variable should be used to set the headless mode. The headless mode property is only required for *UNIX* and it should have a value of -Djava.awt.headless=true
 2. This can be used to pass additional parameters to the *WebLogic Server JVM* when starting the Application Server.

A.4 Set up Database Driver

The version of the *Oracle® Database* driver file, ojdbc6.jar, used by *WebLogic Server* is not the same as that shipped with the application product, under the drivers directory of the *SDEJ* installation.

Follow the step below to set up *WebLogic Server* to use *OracleDatabase Driver* shipped with the application product:

- Replace the ojdbc6.jar file in WLS_HOME\lib folder with one shipped with the application product, under the drivers directory of the *SDEJ* installation, e.g. D:\Curam\SDEJ\drivers;

where, the WLS_HOME variable points to the /server directory of the *WebLogic Server* installation, for example: d:\WLS\weblogic\server or /opt/wls/weblogic/server.

**Note**

The copies of Oracle Thin drivers installed with *WebLogic Server* and other supporting files are installed in WLS_HOME\ext\jdbc\ directory. There is a subdirectory in this folder for each DBMS. If you need to revert to the version of the driver installed with *WebLogic Server* at any point, then you can copy the file from WLS_HOME\ext\jdbc\oracle\11g\ to WLS_HOME\lib folder.

A.5 Starting the WebLogic Administration Service

1. Open a command prompt and navigate to <WebLogic Install directory>/<domain_repository>/<DomainName> directory, (e.g. D:/bea/user_projects/domains/<domain_name>). Execute the command **startWeblogic** from this directory. When the server has started correctly the following line should be displayed: <Server started in RUNNING mode> in the log file, which was configured previously in Section A.3, *Java Virtual Machine Configuration* or in the command prompt if it's not already configured.
2. To open the *Administration Console*, the following URL should be opened in a web browser:
http://<IP Address>:7001/console
3. Login with the username and password configured during installation (e.g. “weblogic”/“weblogic1”).

A.6 Database Configuration

Open the *Administration Console* as detailed in the previous section.

1. Navigate to <DomainName>+Services+JDBC+Data Sources;
2. Click the *New* button;
3. Enter the following fields:
Name: “curamdb”
JNDI Name: “jdbc/curamdb”
Change the *Database Type* to be “Oracle”
4. Click the *Next* button
5. Set the *Database Driver* to be “Oracle’s Driver (Thin XA) for Instance connections; Versions:9.0.1,9.2.0,10,11”;

6. Click the *Next* button
7. Leave the default for *Transaction Options* and click the *Next* button.
8. Set the following fields:

Database Name: This setting depends on how you want to connect to the *Oracle* database; i.e., either using the *Oracle* service name or *Oracle* SID name.

Set this value to the value of `curam.db.oracle.servicename` in `<SERVER_DIR>/project/properties/Bootstrap.properties` to connect to database using the “*Oracle* service name, e.g. .

If you want to connect to an *Oracle* database using the SID name, then set this value to the value of `curam.db.name` in `<SERVER_DIR>/project/properties/Bootstrap.properties` e.g. “yourhost”.

Hostname: Set this value to the value of `curam.db.servername` in `<SERVER_DIR>/project/properties/Bootstrap.properties` , e.g. “gonzo.<host_name>”.

Port: Set this value to the value of `curam.db.serverport` in `<SERVER_DIR>/project/properties/Bootstrap.properties` , e.g. “1521”.

Database Username: Set this value to the value of `curam.db.username` in `<SERVER_DIR>/project/properties/Bootstrap.properties` , e.g. “curam”.

Password: Set this value to the value of `curam.db.password` in `<SERVER_DIR>/project/properties/Bootstrap.properties` , e.g. “p”. Note that password in this property file is encrypted, and you need to set the not encrypted version of this password.

Confirm Password: confirm the entered password.
9. Click the *Next* button
10. Here if you are connecting using *Oracle* service name, then change the *URL* value as shown below before testing the configuration:


```
jd-
bc:oracle:thin:@//serverName:port/databaseServiceName
```

Where *serverName* is the name of the server hosting the database.

Where *port* is the port number the database is listening on.

Where *databaseServiceName* is the service name of the database.

Leave all other fields untouched unless a specific change is required.

Click the *Test Configuration* button to test settings.

11. Click the *Next* button;
12. Review the settings and click the *Next* button;
13. Select *AdminServer* as the target server;



Note

For manual configuration you will need to set the value of `curam.server.name` property in `AppServer.properties` file to be `AdminServer`.

14. Click the *Finish* button;

It is a good idea to restart the *AdminServer* at this point, to ensure the changes are correct. To do this:

1. Navigate to `<DomainName>+Environment+Servers`;
2. Select the *Control* tab, then select *AdminServer* in the Server's list and click *Shutdown+When work completes*; ³
3. Click the *Yes* button to shutdown the *AdminServer*.

A.7 Setup Security

Copy the `CuramMBean.jar` from the `<SDEJ>/lib` directory to the `<WebLogic Server Install directory>/weblogic/server/lib/mbeantypes` directory (`<WebLogic Server Install directory>/wlserver_10.3/server/lib/mbeantypes` for 10.3.3).

Restart the *AdminServer* and start the *Administration Console* as described in the previous section.

1. Navigate to `<DomainName>+Security Realms`;
2. Click on *myrealm* in the *Realms* list;
3. Click on *Providers* tab;
4. Click on *Authentication* tab;
5. Click the *New* button;
6. Enter the following fields:
Name: "myrealmCuramAuthenticator"
Type: "CuramAuthenticator"
7. Click the *OK* button;
8. In the list of *Authentication Providers*, click the *DefaultAuthenticator* checkbox;

9. Click the *Delete* button;
10. Click on *myrealmCuramAuthenticator* in the *Authentication Providers* list;
11. Ensure the *Control Flag* value is set to “REQUIRED”.
If not change the value to “REQUIRED”;
12. Click the *Save* button;
13. Select the *Provider Specific* tab. This tab contains settings for configuring Cúram security in *WebLogic Server*. The defaults should not be changed unless you wish to modify the security configuration. Table A.2, *Provider Specific Options* explains the details of the various options.

You must enter the encrypted password for the Admin Password value. Generate this password by running the supplied utility; e.g. `java -cp %SDEJ%\lib\coreinf.jar curam.util.security.EncryptionAdmin weblogic1 password.txt`.

If any changes are made click the *Save* button;
14. Click the *Save* button; ensure that there are no errors.

Field	Description
Check Identity Only	Optional. If this box is checked the authentication provider will not perform the usual authentication verifications. Instead it will simply ensure that the user exists on the database table. This option is intended where LDAP support is required or an alternative authentication mechanism is to be used.
Admin Username	Required. This is the username of the <i>WebLogic Server</i> administration user. This user is excluded from Cúram authentication.
Admin Password	Required. This is the encrypted password of the <i>WebLogic Server</i> administration user. Generate the encrypted password by running the supplied utility; e.g. <code>java -cp %SDEJ%\lib\coreinf.jar curam.util.security.EncryptionAdmin weblogic1 password.txt</code>
Port	Required. This is the port of the machine on which the Cúram application will run. The default is 7003. In a clustered environment this should be set to a ',' separated list of ports to support multiple servers.
Login Trace	Optional. This box should be checked to debug

Field	Description
	the authentication process. If selected the invocation of the Cúram authentication provider will result in tracing information being added to the <i>WebLogic Server</i> log file.
Run As User	Required. See section Section 4.2.2, <i>Change SYSTEM Username</i> for a description of this property. The default is SYSTEM.
Hostname	Required. This is the hostname of the machine on which the Cúram application will run. The default is localhost. In a clustered environment this should be set to a ',' separated list of host names to support multiple servers.

Table A.2 Provider Specific Options



Note

While configuring the Cúram Authenticator Provider in a clustered environment, the ordering of `hostname` and `port` attributes is important. There is a one to one mapping between the servers and ports specified. For example:

```
Port=7001,7003,7005
```

```
Hostname=host1,host2,host3
```

Here `host1` is running the *WebLogic Server* on port 7001 and `host3` is running the *WebLogic Server* on port 7005

You must enable SSL support. To do this:

1. Navigate to `<DomainName>+Environment+Servers;`
2. Select the *AdminServer* from the list of servers.
3. From the *General* tab click the *SSL Listen Port Enabled* checkbox;
4. Click the *Save* button;
5. Restart your server to take changes.

A.8 Setup JMS

The Cúram application uses persistent messages. A JMS file store must be created for storing persistent messages. To create a directory on the file system where the JMS file store will be kept (e.g. `<WebLogic Server Install directory>/user_projects/domains/<DomainName>/jms_file_store`), complete the following steps using the Administration Console:

1. Navigate to *<DomainName>+Services+Persistent Stores*;
2. Click *New+Create FileStore* and set the following properties:
Name: “CuramJMSFileStore”
Target: “AdminServer”
Directory: <directory created above>;
3. Click the *Finish* (OK for *WebLogic Server 10.3.3*) button;
4. Navigate to *<DomainName>+Services+Messaging+JMS Servers*;
5. Click the *New* button to configure a new *JMSServer* and set the following properties:
Name: “CuramJMSServer”
Persistent Store: “CuramJMSFileStore”
6. Click the *Next* button and select the *AdminServer* as the target server;
7. Click the *Finish* button to complete;

A.9 Queue Setup

1. Navigate to *<DomainName>+Services+Messaging+JMS Modules*;
2. Click the *New* button and enter the following details:
Name: “op-jms”
Descriptor File Name: “jms/op-jms.xml”
3. Click the *Next* button and check the *AdminServer* checkbox as the target server;
4. Click the *Next* button;
5. Click the *Would you like to add resources?* checkbox and click the *Finish* button to complete;

There are four regular queues and two error queues that must be configured. The error queues must be configured first. The following setup should be repeated, replacing *<QueueName>* with each of the following queues (in the order listed): *CuramDeadMessageQueue*, *DPError*, *WorkflowError*, *DPEnactment*, *WorkflowEnactment*, and *WorkflowActivity*.

1. Navigate to *<DomainName>+Services+Messaging+JMS Modules*;
2. Click on *op-jms* in *JMS Modules* list;
3. Click the *New* button inside the *Configuration* tab;
4. Select *Type* “Quota” and click the *Next* button;

5. Enter the following details:
Name: “<QueueName>.Quota”
Leave the default for the *Bytes Maximum*: “9223372036854775807”
Leave the default for the *Messages Maximum*: “9223372036854775807”
Leave the default for the *Policy*: “FIFO”
Leave the default for the *Shared*: “False”.
6. Click the *OK* button;
7. Click the *New* button and select *Type* “Queue”;
8. Click the *Next* button and enter the following details:
Name: “<QueueName>”
JNDI Name: “jms/<QueueName>”.
9. Click the *Next* button;
10. If the *SubDeployments* drop-down list is empty click on *Create a New SubDeployment* and enter *SubDeployment Name*: “CuramJMSServer”;
11. Click the *OK* button;
12. Set *SubDeployment*: “CuramJMSServer”
13. Select “CuramJMSServer” as the target JMS Server;
14. Click the *Finish* button;
15. Click the “<QueueName>” just configured.
16. Click on *Thresholds and Quotas* tab;
17. Set the *Quota*: “<QueueName>.Quota”;
18. Click the *Save* button;
19. Select the *Overrides* tab and set the *Delivery Mode Override* to “Persistent”. Click *Save*;
20. Select the *Delivery Failure* tab and set the *Redelivery Limit* to “1”. Set the *Error Destination* to “none” for “CuramDeadMessageQueue”, “CuramDeadMessageQueue” for “DPErrors” and “WorkflowError”, “DPErrors” for “DPEnactment”, and “WorkflowError” for “WorkflowEnactment” and “WorkflowActivity”.
21. Click the *Save* button;

A.10 Queue Connection Factory Setup

To configure the XA Queue Connection Factory, complete the following

steps:

1. Navigate to *<DomainName>+Services+Messaging+JMS Modules*;
2. Click on *op-jms* in *JMS Modules* list;
3. Click the *New* button inside the *Configuration* tab;
4. Select *Type* “Connection Factory” and click *Next*;
5. Set the following fields:
Name: “CuramQueueConnectionFactory”
JNDIName: “jms/CuramQueueConnectionFactory”;
6. Click the *Next* button;
7. Click the *Finish* button;
8. Click the “CuramQueueConnectionFactory” just configured;
9. Select the *Configuration* tab and then the *Transactions* sub-tab. Ensure that *XAConnection Factory Enabled* is selected. Click the *Save* button.

To configure the non-XA Queue Connection Factory, complete the following steps:

1. Navigate to *<DomainName>+Services+Messaging+JMS Modules*;
2. Click on *op-jms* in *JMS Modules* list;
3. Click the *New* button inside the *Configuration* tab;
4. Select *Type* “Connection Factory” and click *Next*;
5. Set the following fields:
Name: “CuramQueueConnectionFactoryNonXA”
JNDI Name: “jms/CuramQueueConnectionFactoryNonXA”;
6. Click the *Next* button;
7. Click the *Advanced Targeting* button;
8. If the *SubDeployments* drop-down list is empty click on *Create a New SubDeployment* and enter *SubDeployment Name*: “CuramJMSServer”.
9. Click the *OK* button;
10. Set *SubDeployment*: “CuramJMSServer”
11. Select “CuramJMSServer” as the target JMS Server;
12. Click the *Finish* button;
13. Click the “CuramQueueConnectionFactoryNonXA” just configured;
14. Select the *Configuration* tab and then the *Transactions* sub-tab. Ensure

that *XAConnection Factory Enabled* is not selected. Click the *Save* button.

A.11 Topic Setup

To support cache reloading in the Cúram application, a topic must be configured as follows:

1. Navigate to *<DomainName>+Services+Messaging+JMS Modules*;
2. Click on *op-jms* in *JMS Modules* list;
3. Click the *New* button inside the *Configuration* tab;
4. Select *Type* “Quota” and click *Next*;
5. Enter the following details:
 - Name*: “CuramCacheInvalidationTopic.Quota”
 - Bytes Maximum*: Leave the default “9223372036854775807”
 - Messages Maximum*: Leave the default “9223372036854775807”
 - Policy*: Leave the default “FIFO”
 - Shared*: Leave the default “False”.
6. Click the *OK* button,
7. Click the *New* button and select *Type* “Topic”;
8. Click *Next* and enter the following details:
 - Name*: “CuramCacheInvalidationTopic”
 - JNDI Name*: “jms/CuramCacheInvalidationTopic”.
9. Click the *Next* button;
10. If the *SubDeployments* drop-down list is empty click on *Create a New SubDeployment* and enter *SubDeployment Name*: “CuramJMSServer”.
11. Click the *Finish* button;
12. Set *SubDeployment*: “CuramJMSServer”
13. Select “CuramJMSServer” as the target JMS Server;
14. Click the *Finish* button;
15. Click the “CuramCacheInvalidationTopic” just configured.
16. Click on *Thresholds and Quotas* tab;
17. Set the *Quota*: “CuramCacheInvalidationTopic.Quota”;
18. Click the *Save* button;

19. Select the *Delivery Failure* tab and set the *Redelivery Limit* to “1”. Ensure the *Error Destination* is set to (none) and click the *Save* button;

A.12 Topic Connection Factory Setup

1. Navigate to `<DomainName>+Services+Messaging+JMS Modules`;
2. Click on *op-jms* in *JMS Modules* list;
3. Click the *New* button inside the *Configuration* tab;
4. Select *Type* “Connection Factory” and click *Next*;
5. Set the following fields:
Name: “CuramTopicConnectionFactory”
JNDIName: “jms/CuramTopicConnectionFactory”;
6. Click the *Next* button
7. Click the *Advanced Targeting* button;
8. If the *SubDeployments* drop-down list is empty click on *Create a New SubDeployment* and enter *SubDeployment Name*: “CuramJMSServer”.
9. Click the *OK* button;
10. Set *SubDeployment*: “CuramJMSServer”
11. Select “CuramJMSServer” as the target JMS Server;
12. Click the *Finish* button;
13. Click the “CuramTopicConnectionFactory” just configured;
14. Select the *Configuration* tab and then the *Transactions* sub-tab. Ensure that *XAConnection Factory Enabled* is selected. Click the *Save* button.

A.13 Manual Deployment

It is possible to manually deploy an `.ear` file using the *Administration Console*. For this to succeed, the relevant server must be started. In the *Administration Console*, complete the following steps:

1. Navigate to `<DomainName>+Deployments`;
2. Click *Install*.
3. Navigate to the location of the `.ear` file.

The default location for the server `.ear` is:

```
%SERVER_DIR%/build/ear/WLS/Curam.ear
```

4. Select the `.ear` file from the list and click the *Next* button.
5. Accept defaults and click the *Next* button.
6. Accept the defaults and click the *Finish* button.
7. Select the application just deployed in the *Deployments* list and click *Start+Servicing all requests*.
8. Click the *Yes* button;
9. Finally, test the application deployment.

For example, point a Web browser at the *URL* for the deployed application e.g. `https://localhost:7002/Curam`.

Notes

¹The manual steps relate to the *Windows* platform only.

²Refer to the installation guide that is relevant to the platform, i.e. *Windows* or *UNIX*.

³Although it is very easy to kill managed servers by using Ctrl + C when it is running in a command prompt, never do this. Always use the Administration Console to shutdown all managed servers as described above. Using Ctrl + C from a Command Prompt will significantly slow down a machine after a couple of restarts. This is because memory is not released as it should be and the only remedy is to restart the machine.

Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service. IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing

IBM Corporation

North Castle Drive

Armonk, NY 10504-1785

U.S.A.

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

Intellectual Property Licensing

Legal and Intellectual Property Law.

IBM Japan Ltd.

1623-14, Shimotsuruma, Yamato-shi

Kanagawa 242-8502 Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typograph-

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

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

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you. Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Dept F6, Bldg 1
294 Route 100
Somers NY 10589-3216
U.S.A.

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

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

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

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources.

IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products

should be addressed to the suppliers of those products.

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

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

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

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

COPYRIGHT LICENSE:

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

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

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

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

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

Programming Interface Information

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

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/us/en/copytrade.shtml>.

Apache is a trademark of Apache Software Foundation.

Microsoft and Windows 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.

Oracle, WebLogic Server, 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.