

IBM Cúram Social Program Management
Version 7.0.3

*Cúram Deployment Guide for WebSphere
Application Server on z/OS*



Note

Before using this information and the product it supports, read the information in [“Notices” on page 40](#)

Edition

This edition applies to IBM® Cúram Social Program Management v7.0.3 and to all subsequent releases unless otherwise indicated in new editions.

Licensed Materials - Property of IBM.

© **Copyright International Business Machines Corporation 2012, 2018.**

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

© .

Contents

List of Figures.....	v
-----------------------------	----------

List of Tables.....	vi
----------------------------	-----------

Chapter 1. Deploying on IBM WebSphere Application Server on z/OS.....	1
--	----------

Introduction.....	1
Overview.....	1
Assumptions.....	1
Document Conventions.....	1
Third-Party Tools.....	2
Introduction.....	2
Before Installing.....	2
Prerequisites.....	2
Installation.....	2
Post-Installation.....	3
WebSphere Application Server for z/OS.....	3
Prerequisites.....	4
Installation.....	4
Post-Installation.....	4
Apache Ant.....	4
Overview.....	4
Installation.....	4
Post-Installation.....	5
JRE and Java EE.....	5
Overview.....	5
Installation.....	5
Post-Installation.....	5
Building EAR Files.....	5
Introduction.....	5
z/OS-Specific Notes for Building Application EAR Files.....	6
Packaging the Cúram Runtime for Installation on z/OS.....	7
Application Server Configuration.....	8
Introduction.....	8
WebSphere Application Server Configuration.....	8
Security Configuration.....	10
64-bit Mode.....	14
Time Zone Configuration.....	14
Starting and Stopping WebSphere Servers.....	14
Deployment.....	15
Introduction.....	15
Property Files.....	15
Deployment.....	16
Pre-compiling JSPs.....	17
Creating a Database.....	18
Testing Deployment.....	18
Manual WebSphere Application Server Configuration.....	19
Introduction.....	19
Manual WebSphere Application Server Configuration.....	19
Manual Application Deployment.....	37
WebSphere Network Deployment.....	38

Notices.....	40
Privacy Policy considerations.....	41
Trademarks.....	41

List of Figures

1. Sample AppServer properties file.....	9
2. Example of Usage.....	14
3. Example of Usage.....	14
4. Deployment related Bootstrap properties file.....	15
5. Deployment related AppServer properties file.....	16
6. Example of Usage.....	16
7. Example of Usage.....	17
8. Example of Usage.....	18
9. Example shell commands to build a database.....	18

List of Tables

1. z/OS for DB2-specific database properties.....	6
2. Properties dependent on the z/OS filesystem.....	7
3. WebSphere Application Server for z/OS-related port properties.....	7
4. WebSphere Application Server for z/OS structure-related properties.....	7
5. Environment Variables for z/OS UNIX System Services.....	8
6. CuramLoginModule Custom Properties.....	26
7. Exception Destination Settings.....	34

Chapter 1. Deploying on IBM WebSphere Application Server on z/OS

A number of third-party tools are required to deploy IBM Cúram Social Program Management on IBM WebSphere® Application Server for IBM z/OS®. Webclient server and application ear files are required to deploy the application.

Introduction

Overview

This guide describes the process of configuring and deploying IBM Cúram Social Program Management with IBM WebSphere Application Server for IBM z/OS .

The configuration tasks can be summarized as follows:

1. Install and configure required third-party tools;
2. Configure WebSphere Application Server for z/OS for the IBM Cúram Social Program Management application .ear (Enterprise ARchive) files;
3. Build and package the application .ear files.

The .ear files are built separately (on a Microsoft Windows or **UNIX** platform);

4. Deploy the IBM Cúram Social Program Management application and web client, the steps involved are as follows:
 - Establishing property files;
 - Installing the application .ear files;
 - Creating a database;
 - Pre-compiling JSPs (optional);
 - Testing deployment.

WebSphere Application Server for z/OS can be customized and configured in a number of ways for performance, resources, security, and other reasons. This document illustrates a simplistic, single-server approach to configuring WebSphere Application Server for z/OS that may not be appropriate for your installation.

Assumptions

Any team or individual using this document needs to have reasonable knowledge and experience of a wide range of **z/OS** products, technologies, etc. Refer to the *Program Directory for WebSphere Application Server for z/OS 7.0 (GI11-4295)* and related documentation for more information.

The installation and customization of WebSphere Application Server for z/OS, and its related and dependent z/OS -based software is not discussed; however any specific steps required for IBM Cúram Social Program Management are addressed in this document.

Further customer-specific customization may be required, for instance:

- Depending upon your local security (e.g. IBM RACF®) requirements you may have additional configuration and customization to do.

Document Conventions

Several conventions are used in this document:

- Values in angle brackets, e.g. *<WebSphere Configuration Directory>*, refer to substitutions you must provide values for.
- Navigation in the WebSphere Application Server for z/OS Administration Console:
 - "Navigate" refers to selections made via the tree control in the left pane of the browser window and are displayed like this: **Servers > Application Servers**.
 - "Select" refers to hyperlinks that appear in the browser window and are shown in this document as italicized; e.g. *local_host*.
 - "Click" refers to buttons like **OK** and **Next**.
 - "Check" or "Select" refer to check boxes or options that you need to select; e.g. check the **Enforce Java 2 Security** option.

Third-Party Tools

Introduction

To be able to use the IBM Cúram Social Program Management application it is necessary to install and configure software from third parties.

It is beyond the scope of this information to give detailed data and instructions on the installation and configuration of all the various **z/OS** software products needed to support WebSphere Application Server for z/OS and DB2® for z/OS. This information only attempts to give brief details of the required minimum required configuration for each product.

The prerequisites, installation notes and/or post-installation configuration activities are outlined for each of the following:

- DB2 for z/OS;
- WebSphere Application Server for z/OS;
- Apache Ant;
- Java™ SE Runtime Environment (JRE) and Java EE.

Once the third party tools have been installed and configured, the system is ready for the configuration of WebSphere Application Server for z/OS.

Before Installing

In addition to the information in the WebSphere Application Server for z/OS documentation, consider the following items:

- Main storage - Check that your main storage is adequate for running your applications, factoring in the number of users, performance requirements, and so on.
- Filesystem Space - You should plan on allowing additional space in your UNIX System Services filesystem for the Cúram environment and for deployment into the WebSphere Application Server for z/OS configuration.

Prerequisites

Refer to the *Program Directory for IBM DB2 Universal Database for z/OS; version 8 (GI10-8566) and version 9 (GI10-8737)*.

Installation

Before beginning with Cúram configuration and installation it is assumed that DB2 for z/OS has been successfully installed using SMP/E, and the installation has been configured using the ISPF customization panels as per your installation's requirements.

You will need the following information for deploying the application .ear files:

1. Location Name = *<DB2 Location Name>* - specifies your DB2 for z/OS location name. The location name should be displayed in the **z/OS** system log during DB2 for z/OS (DDF) startup:

```

DSNL004I - DDF START COMPLETE
LOCATION <DB2 Location Name>

```
2. User ID = *<database username>* - represents a **z/OS** userid that has all the necessary security access enabled to connect to and manage the DB2 for z/OS database;
3. Password = *<database password>* - is the password for *<database username>*.

Post-Installation

About this task

The following steps may be run using typical DB2 for z/OS interfaces; e.g. SPUFI, DB2 Connect or batch DB2. Supply site-appropriate values to replace those in angle brackets (e.g. *<storage_group>*):

Procedure

1. Create the necessary database Storage Group.

```

CREATE STOGROUP <storage_group> VOLUMES (<volumes>)
VCAT <catalog_name>;

```

2. Create the Cúram application database - the database can be configured for EBCDIC, ASCII, or UNICODE mode, this can be done when creating the database using the CCSID keyword. For ASCII or UNICODE databases see [“Bootstrap properties”](#) on page 6 for information about setting the required property `curam.db.zos.encoding`.

```

CREATE DATABASE CURAM BUFFERPOOL BP0 INDEXBP BP0
STOGROUP <storage_group> CCSID <EBCDIC, ASCII or UNICODE>;

```

3. Ensure that the DSNZPARM RRULOCK parameter, of the DSN6SPRM macro, is set to YES.
4. An environment variable called DB2JCC_LICENSE_CISUZ_JAR must be created in your z/OS UNIX System Services shell environment that points to the installed DB2 for z/OS license jar file used for connectivity to DB2 for z/OS servers on **z/OS**. This is normally named `db2jcc_license_cisuz.jar` and is provided as part of your DB2 for z/OS installation.

Results

Note:

Settings in the DSNZPARM parameter module may need adjustment for compatibility with Cúram. One parameter of particular note is the idle transaction timeout threshold (IDTHT0IN). It may need to be increased for certain CER initialization-related activities following a database build, such as the `prepare.application.data` Ant target as this can run longer than is typically encountered with DB2 for z/OS applications. The time these activities might take and whether they timeout depends on a number of factors, but the symptoms of needing this change may be manifested with the client shell receiving an error similar to:

```

[java] infrastructure:RUN_ID_RUNTIME: A runtime exception occurred:
[jcc][t4][10335][10366][3.63.131] Invalid operation: Connection is closed.
ERRORCODE=-4470, SQLSTATE=08003.

```

Also, in the the z/OS SYSLOG a DSNL027I timeout message with a reason code of 00D3003B may be produced by DB2 in the same timeframe. The relevant DB2 for z/OS documentation provides information on changing the idle thread timeout value.

WebSphere Application Server for z/OS

Prerequisites

Refer to the *Program Directory for WebSphere Application Server for z/OS V7.0 (GI11-4295)* for WebSphere Application Server for z/OS -specific requirements.

Installation

Before beginning with your Cúram configuration and deployment, WebSphere Application Server for z/OS must be installed using the appropriate installation tools as per your site and WebSphere Application Server for z/OS requirements.

WebSphere Application Server for z/OS installation is covered in various IBM publications and in the [WebSphere Application Server, Version V7.0 product documentation](#). However, global security requires further discussion, and this is expanded upon below.

Global Security - Configuring the Security Settings

Turning on WebSphere Application Server for z/OS global security has been described as flipping a big switch and this will impact the behavior of your WebSphere Application Server for z/OS system on z/OS significantly. For this reason it is strongly recommended that you:

- Become familiar with the WebSphere Application Server for z/OS documentation on security. Specifically, you should review:
 - Security topics from the *WebSphere Application Server for z/OS InfoCenter*
 - [IBM WebSphere Application Server for z/OS, Version V7.0: Securing applications and their environment](#)

Be aware that if you have other applications running on WebSphere Application Server for z/OS, they will be impacted by global security being turned on and may no longer function.

Post-Installation

The following step needs to be performed:

- An environment variable called WAS_HOME must be created in your z/OS UNIX System Services shell environment. It should be set to the AppServer directory of the WebSphere Application Server for z/OS installation (e.g. /WebSphere/AppServer).

Apache Ant

Overview

Apache Ant is a Java-based build tool. For those familiar with tools used in other environments it can be viewed as being similar to the make tool.

Installation

The Ant zip file can be obtained from Apache and extracted to a folder on your machine as follows:

- Place the Ant zip file into the z/OS UNIX System Services file system (e.g. /usr/local) and process the file; e.g.:

```
cd /usr/local
jar -xf apache-ant-<version>-bin.zip
```

Where "<version>" represents the appropriate version identified in the *Cúram v6.0 Supported Prerequisites* document.

- Ensure the Ant script in apache-ant-<version>/bin is:
 - In EBCDIC format; e.g.:

```
iconv -t IBM-1047 -f ISO8859-1 apache-ant-<version>/bin/ant \  
> /tmp/ant  
mv /tmp/ant apache-ant-<version>/bin  
– Executable; e.g.:  
chmod a+x apache-ant-<version>/bin/*
```

Post-Installation

About this task

The following steps need to be performed:

Procedure

1. An environment variable called ANT_HOME must be created in your z/OS UNIX System Services shell environment that points to the installation directory chosen for Ant;
2. Add \$ANT_HOME/bin to the execution path via your PATH z/OS UNIX System Services environment variable;
3. Create a system environment variable, ANT_OPTS, in your z/OS UNIX System Services shell environment that should be set to be at least-Xmx512m.

Results

Test Ant by running:

```
ant -version
```

You should see output indicating the version and compilation date of Ant.

JRE and Java EE

Overview

Both the JRE and Java EE are necessary.

Installation

Specific installation instructions are not provided for the JRE and Java EE on **z/OS** as WebSphere Application Server for z/OS, Version 7.0 provides an integrated JRE and Java EE, which must be used. See the appropriate IBM-supplied information for your particular environment.

Post-Installation

- An environment variable called JAVA_HOME must be created in your z/OS UNIX System Services shell environment that points to the installed JRE. \$JAVA_HOME should be set to \$WAS_HOME/java. \$JAVA_HOME/bin should be placed in the path via your \$PATH environment variable.
- An environment variable called J2EE_JAR must be created in your z/OS UNIX System Services shell environment that points to the installed Java EE jar file. This should point at \$WAS_HOME/lib/j2ee.jar.

Building EAR Files

Introduction

The main step before deployment of an IBM Cúram Social Program Management is to package the application into EAR (Enterprise ARchive) files. However, the building of the application .ear files cannot

be done on **z/OS** and must be done on Windows or any other environment that is identified as a being supported for building in the *Cúram Supported Prerequisites* document.

The remainder of this chapter outlines **z/OS** -specific requirements for building **z/OS** -compatible .ear files. You may also find helpful information from the following manuals:

- *Cúram Application Workshop Guide* - This manual has basic build instructions for application .ear files;
- *Cúram Server Developer's Guide* - This manual has detailed instructions for a server build (chapter 3);
- *Cúram Web Client Reference Manual* - This manual has detailed instructions for web client development including installation and configuration (chapter 4);

z/OS-Specific Notes for Building Application EAR Files

These sections highlight specifics for building **z/OS** -compatible .ear files.

Property Files

When building an IBM Cúram Social Program Management application the `Bootstrap.properties` and `AppServer.properties` files must be set correctly for the target **z/OS** platform.

Bootstrap properties

The `Bootstrap.properties` file contains the machine-specific configuration properties for initially getting a connection to the database. Pay specific attention to the following elements:

1. database properties:

Table 1: z/OS for DB2-specific database properties	
Property	Notes
curam.db.type	Value must be set to "zos".
curam.db.zos.enableforeignkeys	Set appropriately for your environment ("true" or "false").
curam.db.zos.encoding	Specifies whether the database being used on z/OS requires processing for EBCDIC, ASCII, or UNICODE. This should be set to "EBCDIC", "ASCII", or "UNICODE" depending on the appropriate database encoding in use. "EBCDIC" is the default value.
curam.db.zos.dbname	Value must be the name of the DB2 for z/OS database.
curam.db.zos.32ktablespace	Value must be the name of the DB2 for z/OS 32K tablespace.
curam.db.username	Value depends on the configuration of your z/OS system.
curam.db.password	Value depends on the configuration of your z/OS system. Since this is an encrypted password you must generate it by running the Ant encrypt target on any supported platform; e.g. <code>cd \$CURAMSDEJ/bin; ant encrypt -Dpassword=<The password for curam.db.username></code>
curam.db.name	Value is the DB2 for z/OS location name.
curam.db.servername	Value depends on the hostname (or IP address) of your DB2 for z/OS system.

Table 1: z/OS for DB2-specific database properties (continued)	
Property	Notes
curam.db.serverport	Value depends on the configuration of your DB2 for z/OS system.

2. filesystem-dependent properties:

Table 2: Properties dependent on the z/OS filesystem	
Property	Notes
curam.environment.bindings.location	Value must reflect a valid directory in the target z/OS UNIX System Services filesystem.

AppServer properties

Pay specific attention to the following elements:

1. WebSphere Application Server for z/OS port-related properties are shown in [Table 3 on page 7](#).

Table 3: WebSphere Application Server for z/OS-related port properties	
Property	Notes
curam.server.port	Value must match the WebSphere Application Server for z/OS bootstrap port (see “Set up the Port Access” on page 29).
curam.client.httpport	Value must match the CuramClientEndPoint port value (see “Set up the Port Access” on page 29).
curam.webservices.httpport	Value must match the CuramWebServicesEndPoint port value (see “Set up the Port Access” on page 29).

2. WebSphere Application Server for z/OS structure-related properties are shown in [Table 4 on page 7](#).

Table 4: WebSphere Application Server for z/OS structure-related properties	
Property	Notes
curam.server.host	Value depends on the hostname (or IP address) of your DB2 for z/OS system.
curam.server.name	Value must match the name of the target WebSphere Application Server for z/OS server.
cell.name	Value must match the name of the target WebSphere Application Server for z/OS cell.
node.name	Value must match the name of the target WebSphere Application Server for z/OS node.
profile.name	For WebSphere Application Server for z/OS the only profile name supported is "default", which is the default.

Packaging the Cúram Runtime for Installation on z/OS

After you have built the .ear files you must package them and the runtime environment for installation on **z/OS**.

For example, on **Windows** (with your environment setup as per the *Cúram Deployment Guide for WebSphere Application Server*) enter the following commands:

```
cd %SERVER_DIR%
build release
jar -cf release.zip release
```

You must then FTP or copy the `release.zip` file to your target **z/OS** filesystem location.

To unzip the `release.zip` file on **z/OS** you should establish two environment variables in your z/OS UNIX System Services shell environment for this task and subsequent tasks:

Table 5: Environment Variables for z/OS UNIX System Services	
Environment Variable	Value
SERVER_DIR	represents the location you will unzip the <code>release.zip</code> into; e.g.: <code>/curam/release</code> .
CURAMSDEJ	represents the directory for running build scripts: <code>\$CuramSDEJ</code> .

With the `release.zip` on your **z/OS** system, in your shell environment enter the following commands to unzip it:

```
mkdir -p $SERVER_DIR
cd $SERVER_DIR/..
jar -xf <from FTPed location>/release.zip
```

Application Server Configuration

Introduction

This chapter presumes that WebSphere Application Server for z/OS has already been installed on **z/OS**. Consult [“Third-Party Tools” on page 2](#) for Cúram -specific information on the installation of WebSphere Application Server for z/OS.

The configuration of WebSphere is similar on all platforms and a number of Ant targets are available to aid the configuration and management of the installation. For those interested, [“Manual WebSphere Application Server Configuration” on page 19](#) details the manual steps performed by the configuration scripts.

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

Note: On WebSphere Application Server for z/OS the only profile available is the *default* profile, no other option is possible.

The `configure` target uses the *default* profile created by WebSphere Application Server for z/OS. It is strongly recommended that you have a backup copy of your WebSphere Application Server for z/OS configuration filesystem in case you need to rerun the `configure` target for any reason.

WebSphere Application Server Configuration

The configuration of WebSphere Application Server for z/OS involves setting up a data source, a number of servers and configuring the JMS and security settings. All these tasks can be performed by executing the provided `configure` target.

The profile created by the Ant `configure` target will take the following defaults. When calling the target the `cell.name` property may be overridden; however, the `profile.name` property may not have any value other than "default" because that is the only value supported by WebSphere Application Server for z/OS.

- `profile.name=default`
- `cell.name=${node.name}Cell`

The command `build.sh 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` directory. It is possible to override this default location for the properties file by specifying `-Dprop.file.location=<new location>` when executing the configure target. See [“Property Files”](#) on page 6, and the *Cúram Server Developer's Guide* for more information on the setup of a `Bootstrap.properties`. [“WebSphere Application Server Configuration”](#) on page 8, shows example contents of the `AppServer.properties` file.

By default the configure target establishes a DB2 Universal Type 4 Driver (XA) data source. However, you may configure a DB2 Universal Type 2 Driver (RRS) data source by setting the `curam.db.type2.required` property in `AppServer.properties`. When using this property you must have the `DB2DIR` environment variable set to your DB2 for z/OS installation path.

There are a number of possible ways of configuring DB2 for z/OS and WebSphere Application Server for z/OS to support a Type 2 driver. Review the [WebSphere Application Server, Version 7.0 product documentation](#), including the article ["DB2 Universal JDBC Driver Support"](#), and related information.

It is possible to configure a Type 2 Universal Driver by passing an optional property `curam.db.zos.jcc.propfile`, specifying the fully qualified name of a DB2 for z/OS jcc property file that will be set in the servant JVM `db2.jcc.propertiesFile` property, which may contain various settings such as the subsystem ID.

```
### APPLICATION SERVER PROPERTIES

# Property to indicate WebSphere is installed.
as.vendor=IBM

# The username and encrypted password for admin server.
security.username=<e.g. websphere>
security.password=<encrypted password>

# The name of the WebSphere Cell.
cell.name=mycell

# The name of the WebSphere Node.
node.name=MyNode

# The name of the server on which the application will be hosted.
curam.server.name=CuramServer
curam.server.port=2809

# The alias that should be used for the database authorization
curam.db.auth.alias=dbadmin

# HTTP Port for the server on which the client
# will be accessed
curam.client.httpport=9044

# HTTP Port for the server on which the Web services
# will be accessed
curam.webservices.httpport=9082

# Property to set JVM initial and maximum heap size.
curam.server.jvm.heap.size=1024
```

Figure 1: Sample AppServer properties file

By default the configure target sets the JVM initial and maximum heap size to "1024" MB. However, you can override the default JVM initial and maximum heap size by setting the `curam.server.jvm.heap.size` property in the `AppServer.properties` file.

For WebSphere Application Server for z/OS you must also include a property `cell.name` that is equal to the long name of the cell.

Note:

1. The setting of the Java heap as described in the [“WebSphere Application Server Configuration”](#) on page 8 example and set by the configuration scripts is for illustrative purposes. Based on the size of your customized application, deployment strategy, etc. these settings may be too low or too high. The optimum value should be determined by monitoring the memory performance of your server.

2. Memory issues may occur with the WebSphere Application Server for z/OS wrapped database drivers during the retrieval of large CLOBs and BLOBs (3MB+) from the database. These issues may be worked-around by increasing the Max Heap Size JVM parameter as appropriate on the deployed server.

Alternative JAR File Locations

If the WebSphere Application Server for z/OS configuration file system is mounted as read-only, the placement of the Cúram registry and cryptography JAR files will fail. By default, each time that the Ant configure target is run, the JAR files are copied into the WebSphere configuration file system at \$JAVA_HOME/lib/ext and \$WAS_HOME/lib. If the underlying installation file system is mounted as read-only, the copy of the JAR files will fail. It is not feasible to remount the file system as read/write for each configure invocation. However, it is possible to set up a symbolic link with the file system mounted as read/write, which is a one-time activity, and to then specify an alternative location for the files to be copied to.

The following steps show the one-time procedure:

1. Mount the WebSphere installation file system, for example, /usr/lpp/zWebSphere/, as read/write.
2. Create a symbolic link in the WebSphere lib directory for the Cúram Registry.jar file, which contains the CuramLoginModule, as shown in the following example:

```
ln -s /curam/EJBServer/CuramSDEJ/lib/Registry.jar /usr/lpp/zWebSphere/lib/Registry.jar
```

3. Create a symbolic link in the Java lib/ext directory for the Cúram cryptography CryptoConfig.jar file, as shown in the following example:

```
ln -s /curam/EJBServer/project/properties/CryptoConfig.jar /usr/lpp/zWebSphere/java/lib/ext/CryptoConfig.jar
```

4. Remount the WebSphere installation file system as read-only.

The previous steps enable your WebSphere file system to remain read-only when the configure target is run, which copies the files to the alternative location that points to the installation file system. When you run the Ant configure target, specify the properties that are shown in the following code sample, which uses the example locations from the previous steps:

```
-Dcrypto.ext.dir=/curam/EJBServer/project/properties/  
-Dregistry.jar.file.location=/curam/EJBServer/CuramSDEJ/lib/
```

Security Configuration

The default security configuration of IBM Cúram Social Program Management within WebSphere Application Server for z/OS involves the default file-based user registry and a JAAS Login Module. The *Default Configuration for IBM WebSphere Application Server* section in the *Cúram Security Handbook* should be referenced for further details on this.

There are a number of alternative security configurations that can be used with WebSphere Application Server for z/OS. The configurations are available to support the use of alternative authentication mechanisms, such as an LDAP directory server or a single sign-on solution.

To avail of a different configuration the properties detailed in the following sections should be set in the AppServer.properties file before running the configure target. Any alternative authentication mechanisms should be configured manually after running the configure target with the relevant properties set. To configure the login module for identity only authentication the curam.security.check.identity.only property should be set to true. This is to ensure that the configured alternative authentication mechanism is used.

The Identity Only Authentication section in the *Cúram Security Handbook* should be consulted for further details.

SAF (RACF) Configuration

When configuring your WebSphere Application Server for z/OS system to use SAF (RACF), having configured WebSphere Application Server for z/OS appropriately with the z/OS Profile Management Tool or ISPF customization panels, you must set the `curam.security.zos.saf` property to true before running the `configure` target.

When running the `configure` target the default value for property `curam.security.user.registry.enabled` is true. Overriding `curam.security.user.registry.enabled` by setting it to false is not recommended. Property `curam.security.check.identity.only` can be set as per your requirements (see below).

Special Configuration Steps When Using Identity Only and LDAP

About this task

When using identity only in combination with WebSphere Application Server for z/OS and LDAP you may need to perform additional manual configuration steps; this is regardless of whether configuration is done via the WebSphere Application Server for z/OS Administrative Console or the `configure` target. With this combination you may find that WebSphere Application Server for z/OS fails to start successfully and this is due to the need to add a WebSphere Application Server for z/OS -generated username to the login module `exclude` list property (`exclude_usernames`) described in “Add the Login Module” on page 26. In this case of WebSphere Application Server for z/OS failing to start there will be a SECJ0270E error message in the `SystemOut.log` file prior to the failure.

These are the steps needed to resolve this error:

Procedure

1. Identify the username that is causing WebSphere Application Server for z/OS start to fail. Configure the login module trace as described in “Logging the Authentication Process” on page 12 (in regard to the `configure` target) or “Add the Login Module” on page 26 (in regard to configuring via the Administrative Console), and restart WebSphere Application Server for z/OS. With the login module trace running, prior to the SECJ0270E error in the `SystemOut.log` file, the trace data will identify the failing username with a record like this:

```
SystemOut      0 Username: server:MyNodeCell_MyNode_CuramServer
```

Where "MyNode" is the node name, "MyNodeCell" is the cell name, and "CuramServer" is the WebSphere Application Server for z/OS server name. Following the login module trace data will be the error, which will look like this:

```
SECJ0270E: Failed to get actual credentials.  
The exception is javax.security.auth.login.LoginException:  
Context: MyNodeCell/nodes/MyNode/servers/CuramServer,  
name: curamejb/LoginHome:  
First component in name curamejb/LoginHome not found.
```

2. Specify the failing username in the login module `exclude_usernames` property in the WebSphere Application Server for z/OS configuration. Since WebSphere Application Server for z/OS is failing to start you cannot make this change via the Administrative Console and you must edit the WebSphere Application Server for z/OS configuration file directly. In the WebSphere Application Server for z/OS configuration file system edit `config\cells\MyNodeCell\security.xml`, which will have three occurrences of the `exclude_usernames` property (one for each alias); e.g.:

```
<options xmi:id="Property_1301940482165"  
  name="exclude_usernames"  
  value="websphere,db2admin"  
  required="false"/>
```

You must modify the three occurrences to include the newly identified username from the trace entry above; e.g.:

```
<options xmi:id="Property_1301940482165"  
  name="exclude_usernames"  
  value="websphere,db2admin,server:MyNodeCell_MyNode_CuramServer"  
  required="false"/>
```

Note that in the `exclude_usernames` occurrences the `id` attribute will vary per your system configuration and the comma separator in the example value attribute represents the default `curam.security.usernames.delimiter` value, which may be different in your case.

3. Restart WebSphere Application Server for z/OS.

WebSphere Application Server User Registry

By default the configured WebSphere Application Server for z/OS user registry is not queried as part of authentication. When the login module is configured for identity only, the user registry is queried. It is possible to override this default behavior by setting the `curam.security.user.registry.enabled` property. If this property is set to `true` the WebSphere Application Server for z/OS user registry will be queried during the authentication process, regardless of whether identity only authentication is enabled or disabled. If this property is set to `false`, the WebSphere Application Server for z/OS user registry will not be queried. For example, if `curam.security.check.identity.only` is set `true` and `curam.security.user.registry.enabled` is set to `false`, neither the Cúram authentication verifications nor the WebSphere Application Server for z/OS user registry will be used as part of the authentication process.

You can also control the authentication of types of external users (i.e. non-internal users) against the WebSphere Application Server for z/OS user registry via use of the `curam.security.user.registry.enabled.types` and/or the `curam.security.user.registry.disabled.types` properties. These properties specify a comma-delimited list of external user types that will, or will not be, authenticated via the WebSphere Application Server for z/OS user registry:

- User types specified in the `curam.security.user.registry.enabled.types` list will be processed against the WebSphere Application Server for z/OS user registry (e.g. LDAP) and your `ExternalAccessSecurity` implementation.
- User types specified in the `curam.security.user.registry.disabled.types` list will not be processed against the WebSphere Application Server for z/OS user registry and the processing of your `ExternalAccessSecurity` implementation will be the authority for authentication.

The precedence order in processing these three properties and the WebSphere Application Server for z/OS user or external (e.g. LDAP) registry is as follows:

- By default the WebSphere Application Server for z/OS user registry is not checked and the application authentication is used.
- The setting of the `curam.security.user.registry.enabled` property to `true` requires authentication by both the WebSphere Application Server for z/OS, or external (e.g. LDAP), user registry and application security (for internal users) or your `ExternalAccessSecurity` implementation (for external users).
- An external user of a type specified in the `curam.security.user.registry.enabled.types` list must be authenticated by the WebSphere Application Server for z/OS, or external, user registry and your `ExternalAccessSecurity` implementation.
- An external user of a type specified in the `curam.security.user.registry.disabled.types` list is not authenticated by the WebSphere Application Server for z/OS, or external, user registry and your `ExternalAccessSecurity` implementation is the authority.

See [“Set up the System JAAS Login Module” on page 26](#) for more information on setting the resultant properties in the `CuramLoginModule` configuration.

Logging the Authentication Process

`curam.security.login.trace` is an optional property that will enable logging for the login module. When set to `true` this property results in tracing information being added to the WebSphere Application Server for z/OS `SystemOut.log` file during the authentication process.

Establishing an Alternate Exclude Username Delimiter

`curam.security.usernames.delimiter` is an optional property that will enable setting an alternate delimiter for the list of usernames in the `exclude_usernames` property. The property can be set to a character that will allow usernames with embedded commas such as with LDAP.

WebSphere Application Server Caching Behavior

WebSphere Application Server for z/OS caches user information and credentials in a security cache and the application login module will not be invoked while a user entry is valid in this cache. The default invalidation time for this security cache is ten minutes, where the user has been inactive for ten minutes. The *WebSphere Caching Behavior* section in the *Cúram Security Handbook* should be consulted for further details on this.

Security custom properties

- `com.ibm.ws.security.webChallengeIfCustomSubjectNotFound`

This property determines the behavior of a single sign-on LTPA Token2 login.

When this property value is set to true, the token contains a custom cache key, and the custom Subject cannot be found, the token is used to log in directly as the custom information needs to be gathered again. A challenge occurs so that the user to login again. When this property value is set to false and the custom Subject is not found, the LTPA Token2 is used to login and gather all of the registry attributes. However, the token might not obtain any of the special attributes that downstream applications might expect.

By default the configuration script sets a WebSphere Application Server for z/OS property, `com.ibm.ws.security.webChallengeIfCustomSubjectNotFound`, to `false` to ensure that web sessions can seamlessly transfer between two servers in a cluster (for example, in a fail over scenario) without being asked for security credentials. This setting allows the security token used by WebSphere Application Server for z/OS to be validated correctly, without user input.

If this behavior is not required it is possible to change this property to true, see [“Set up the System JAAS Login Module” on page 26](#) for more information on setting *Security custom properties*. If the property is set to `true`, when a web session switches from one server in the cluster to another, perhaps due to the original server failing, the user will be asked for security information before being able to proceed.

Security hardening measures

When a user logs into the application, they provide a username & password. This is sent to the server, and if successfully authenticated, the server responds with a unique token. The token, in this case, is 'LTPA token'. This token is used in all subsequent requests to recognize the user and then serves privileged content. When the user logs out, we would expect this token to become invalid. but this is not the case and there is no way to invalidate the LTPA token, which has been confirmed by IBM. **IBM's recommendation is to use two "security hardening measures" of:**

1. Setting the security Requires SSL option;
2. Setting a custom property to limit LTPA cookies to SSL only.

The default configuration scripts make this change and the steps are documented [“Configure Administration Security” on page 24](#).

For more information see:

- http://www.ibm.com/developerworks/websphere/techjournal/1004_botzum/1004_botzum.html?ca=drs#step19
- http://www.ibm.com/developerworks/websphere/techjournal/1004_botzum/1004_botzum.html?ca=drs#step29

Cúram Cryptography

Cúram cryptography relates to functionality for managing passwords and is covered in detail in the *Cúram Security Handbook* and you should consult it in consideration that:

- For production environments it is strongly recommended that you modify the default settings.
- For development and test environments you need to consider where the defaults provide acceptable protection in your environment.

- For users upgrading from a previous version of IBM Cúram Social Program Management, existing passwords will not work out-of-the-box. You can, if you are willing to accept a reduced level of security, at your own risk, choose to take the steps to leave existing system and user passwords as-is, but this is not recommended. More information about upgrading is available in the *Cúram Upgrade Guide*.

64-bit Mode

If using the configure target the `curam.zos.64bitmode` property can be specified in the `AppServer.properties` file with a value of `true` to have the server configured for 64-bit mode support.

Note: When using 64-bit mode support you may also need to review and adjust your JVM heap sizes based on your application size, throughput, performance goals, and other factors.

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.

Starting and Stopping WebSphere Servers

A number of Ant targets are provided to aid in the starting and stopping of WebSphere Application Server for z/OS servers. These targets should be executed from the `<SERVER_DIR>` directory and as for the configure target, they require the `AppServer.properties` file to be setup correctly ([“WebSphere Application Server Configuration” on page 8](#)). They also require a number of extra parameters to be specified and these are detailed below.

Start a WebSphere Server

The Ant target for starting a WebSphere Application Server for z/OS server is `startserver` and requires the following options:

- `-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 `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 `prepare.application.data` target must also be rerun.

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

Figure 2: Example of Usage

Stop a WebSphere Server

The Ant target for stopping a WebSphere Application Server for z/OS server is `stopserver` and requires the following options:

- `-Dserver.name`

The name of the server to be stopped.

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

Figure 3: Example of Usage

Restart a WebSphere Server

The Ant target for restarting a WebSphere Application Server for z/OS server is `restartserver` and the options are the same as for the `startserver` target. See [“Start a WebSphere Server” on page 14](#) for an example of usage.

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

Deployment

Introduction

The final step, after packaging the IBM Cúram Social Program Management application and web services application in `.ear` files and configuring WebSphere Application Server for z/OS, is to deploy the `.ear` files to the application server.

Before deploying, it is important to note that in WebSphere Application Server for z/OS the configuration scripts provided with IBM Cúram Social Program Management support a simple configuration targeted at a base server installation of WebSphere Application Server for z/OS.

Deployment involves:

- Establishing property files;
- Installing the `.ear` files;
- Creating a database;
- Optionally, but strongly recommended, pre-compiling the JSPs;
- Testing the application.

Property Files

To install application `.ear` files using Ant you must have appropriate property files in your `$SERVER_DIR/project/property` directory. These files are:

- `Bootstrap.properties` - for creating a database;
- `AppServer.properties` - for installing `.ear` files.

This section outlines what these files need to contain. For more information see the *Cúram Server Developer's Guide*.

Bootstrap.properties

Specific or relevant deployment properties for WebSphere Application Server for z/OS are shown in [“Bootstrap.properties” on page 15](#).

```
# DATABASE-SPECIFIC (DB2 for z/OS)
curam.db.type=ZOS
curam.db.zos.encoding=EBCDIC
curam.db.zos.enableforeignkeys=false
curam.environment.bindings.location=
  /<Value of $SERVER_DIR>/project/properties

curam.db.username=<database username>
curam.db.password=<encrypted database password>

curam.db.name=<DB2 Location Name>
curam.db.servername=<host name>
curam.db.serverport=<DB2 port>

curam.db.zos.dbname=CURAM
curam.db.zos.32ktablespace=CURAMTS
```

Figure 4: Deployment related Bootstrap properties file

Some of these properties are described in [“Bootstrap properties”](#) on page 6 and are the same as what you need for building IBM Cúram Social Program Management on Windows to deploy to z/OS, but note the following:

- The `<Value of $SERVER_DIR>` is the value of your `$SERVER_DIR` environment variable.

AppServer.properties

Specific or relevant deployment properties for WebSphere Application Server for z/OS are shown in [“AppServer.properties”](#) on page 16.

```
# Property to indicate WebSphere
as.vendor=IBM

# The name of the WebSphere Cell.
cell.name=mycell

# The name of the WebSphere Node.
node.name=mynode

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

Figure 5: Deployment related AppServer properties file

Some of these properties are described in [“AppServer properties”](#) on page 7 and are the same as what you need for building IBM Cúram Social Program Management application .ear files to deploy on z/OS.

Checking the Configuration

You can check your property files and configuration by running the Ant `configtest` target.

Run the `configtest` target from the shell as follows:

```
cd $CURAMSDEJ/bin
ant configtest
```

Review the output for any errors or warnings and resolve them.

Deployment

There are Ant targets for installing and un-installing applications on a WebSphere Application Server for z/OS server. As with the `startserver` and `stopserver` targets, the `installapp` and `uninstallapp` targets require that the `AppServer.properties` file is configured correctly (see [“WebSphere Application Server Configuration”](#) on page 8). The targets also require a number of options to be specified and these are detailed below.

Ensure the server is started before installing an application. There is no need to restart the server after installation, as the `install` target will automatically start the application.

Install an Application

The Ant target to install an application (in the form of an .ear file) is `installapp` and requires the following options:

- `-Dserver.name`
The name of the server to install the application.
- `-Dear.file`
The fully qualified name of the .ear file to install.
- `-Dapplication.name`
The name of the application.

```
build.sh installapp -Dserver.name=CuramServer
-Dear.file=/ear/Curam.ear
-Dapplication.name=Curam
```

Figure 6: Example of Usage

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

An optional Ant property is available for passing additional arguments to WebSphere wsadmin: `wsadmin.extra.args`. For example, the following sets new Java heap sizes and passes the option to append wsadmin tracing:

```
-Dwsadmin.extra.args="-javaoption -Xms1024m -javaoption -Xmx1024m -appendtrace true"
```

Depending on your shell you may need to escape the quotes above; e.g. `-Dwsadmin.extra.args=\"-appendtrace true\"`. You should not use this property to set arguments already passed via the Curam Ant scripts and you can observe these when running Ant by specifying its verbose option: `-v`.

Change SYSTEM Username

It is strongly recommended that you change the username for JMS invocation while deploying the application. The following properties should be set in the `AppServer.properties` file before deployment to modify this username:

- `curam.security.credentials.async.username`

The username JMS invocations should run under.

- `curam.security.credentials.async.password`

The encrypted password associated with the username. The password must be encrypted using the Ant `encrypt` target. See the *Cúram Server Developers Guide* for more information.

It is also possible to change the username once the application has been deployed using the WebSphere Application Server for z/OS Administrative Console. Navigate to **Applications > Application Types > WebSphere enterprise applications** and select the application. Select the **User RunAs roles** link. Check the **everyone** role, enter a new username and password (note, password should be entered in the unencrypted format here) and click the **Apply** button. Save the changes as detailed in [“Save the Master Configuration” on page 24](#).

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.

Uninstall an Application

The Ant target to uninstall an application is `uninstall` 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.sh uninstallApp -Dserver.name=CuramServer  
-Dapplication.name=Curam
```

Figure 7: Example of Usage

Pre-compiling JSPs

There is one additional target available during deployment, `precompilejsp`, which allows for the JSPs of a client .ear to be pre-compiled *before* installing the .ear file. Pre-compiling the JSPs before installation will speed 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.sh precompilejsp -Dear.file=$SERVER_DIR/ear/WAS/Curam.ear
```

Figure 8: Example of Usage

Note: This is a long running activity and depending upon the capabilities of your system, etc. could take several hours. Ensure your task is not significantly restricted with respect to available CPU time and that there is adequate free space available in the \$CURAMSDEJ file system.

Also while running the `precompilejsp` target for WebSphere Application Server for z/OS, an out of memory exception may occur (or some JSPs may silently be ignored and not pre-compiled). To work around this the `JspBatchCompiler.sh` script in the `$WAS_HOME/bin` directory should be modified to increase the maximum memory size. Change the memory consumption from `-Xmx256m` to at least `-Xmx1024m`.

Creating a Database

To use the IBM Cúram Social Program Management application you must create and initialize a database. This section assumes you are using the Ant database target to create a database. However, it is possible to use DB2 client tools to do this. See the *Cúram Installation Guide* for more details on this method.

```
cd $CURAMSDEJ/bin
ant database
```

Figure 9: Example shell commands to build a database

Testing Deployment

When the IBM Cúram Social Program Management application .ear file(s) is installed on a configured WebSphere Application Server for z/OS installation the next step is to start and test the application.

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

Ensure the relevant server is started 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 WebSphere Application Server for z/OS system is running, `<port>` identifies the server port on which client application is deployed (as in “Set up the Port Access” on page 29) and `<context-root>` identifies the Context Root of the WAR module. There is no need to restart the server after an application is deployed.

Before the page can be opened, the browser will be directed to the login page. Login with a valid Cúram 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.

Using IBM WebSphere Application Server with USGCB

The United States Government Configuration Baseline (USGCB) is a federal government-wide initiative that provides guidance to agencies on what should be done to improve configuration settings, focusing mainly on security. When running the IBM Cúram Social Program Management application, if using IBM WebSphere Application Server v7, (see the *IBM Cúram Social Program Management Supported Prerequisites* guide for supported versions of IBM WebSphere Application Server v7) with USGCB settings, it's possible that images may be missing. If this issue occurs, it indicates that IBM WebSphere Application Server does not recognize .png files. To fix this issue, IBM WebSphere Application Server must be updated to support the PNG MIME type. For details on this, the *WebSphere Application Server Information Center* documentation should be consulted.

For more information on USGCB, see <http://usgcb.nist.gov/>

Manual WebSphere Application Server Configuration

Introduction

This covers the manual steps required to configure and deploy on a base application server installation of WebSphere Application Server for z/OS. You will have to alter these steps appropriately to deploy in a Network Deployment installation of WebSphere Application Server for z/OS. See [“WebSphere Network Deployment”](#) on page 38 for more information in this area.

Manual WebSphere Application Server Configuration

The IBM WebSphere Application Server for z/OS installation can be configured manually if required, but this is not recommended when using a base application server installation. This section details the manual steps required to configure WebSphere Application Server for z/OS for information purposes only.

It is worth noting that any settings entered under the **Resources** section of the Administrative Console can be configured at multiple levels that control the JNDI scope. These include cell, node, or server. Upon selecting a **Resource**, the top of the main browser window shows this scope and allows the various resources in the current scope to be viewed. The scope, and in turn the location of any resources set, should be based upon planned use, i.e. if working in a cluster it may not be necessary to set the same settings on each server, so the scope may be set to cell or node.

The Administrative Console

Most of the configuration of WebSphere Application Server for z/OS is done using the WebSphere Administrative Console. To run the Administrative Console the server in the default profile must be started, since the Administrative Console is installed as a web application on this server (see [“Starting and Stopping WebSphere Servers”](#) on page 14 for more information on starting servers).

To open the Administrative Console, a web browser should be pointed at:

`http://<Your WebSphere host>:<protocol_http_port>/ibm/console`

Where:

<Your WebSphere host> identifies the host name or IP address where your WebSphere Application Server for z/OS system is running and *<protocol_http_port>* identifies the port assigned in your installation and customization of WebSphere Application Server for z/OS.

Scripting Support

To support the execution of provided Ant scripts it is necessary to change the WebSphere Application Server for z/OS property files.

sas.client.props

Open the `sas.client.props` file found in the `profiles/default/properties` directory of the WebSphere Application Server for z/OS installation. It is necessary to set the login source to retrieve the username and password from a properties file rather than having to type them in each time the scripts are run. Set or where necessary add the following properties:

```
com.ibm.CORBA.loginSource=properties
# RMI/IIOP user identity
com.ibm.CORBA.loginUserId=websphere
com.ibm.CORBA.loginPassword=websphere
```

where *websphere* is the username and password for the Administrative Console.

soap.client.props

Open the `soap.client.props` file, also found in the `profiles/default/properties` directory the WebSphere Application Server for z/OS installation. It is necessary to set the login source to retrieve the username and password from a properties file rather than having to type them in each time the scripts are run. Set the following properties to match the credentials you configured for WebSphere as in

“WebSphere Application Server Configuration” on page 8. In the example below the values are merely examples and the password specified in this file cannot be encrypted:

```
com.ibm.SOAP.loginUserId=websphere
com.ibm.SOAP.loginPassword=websphere
```

where *websphere* is the username and password for the Administrative Console.

To avoid timeouts when installing application .ear files ensure that the following is set appropriately; e.g.:

```
com.ibm.SOAP.requestTimeout=3600
```

Depending on the performance of your environment you may need a different value.

server.policy

Open the `server.policy` file found in the `profiles/default/properties` directory of the WebSphere Application Server for z/OS installation. Add the following lines to the end of this file:

```
grant codeBase "file:<CURAMSDEJ>/drivers/-" {
permission java.security.AllPermission;
};
```

where `<CURAMSDEJ>` is the SDEJ installation directory.

```
grant codeBase "file:${was.install.root}/
profiles/default/installedApps/
<cell.name>/<SERVER_MODEL_NAME>.ear/
guice-2.0.jar" { permission java.lang.RuntimePermission
"modifyThread"; permission java.lang.RuntimePermission
"modifyThreadGroup"; };
```

where `<cell.name>` is the name of the target WebSphere Application Server for z/OS cell

and `<SERVER_MODEL_NAME>` is the name of the application .ear (EAR) file.

Creating the Data Source Login Alias

About this task

DB2 for z/OS is the database supported on z/OS. The WebSphere Application Server for z/OS Administrative Console is used to configure a login alias for the DB2 for z/OS data sources as follows:

Procedure

1. Navigate to **Security > Global security**;
2. Expand the **Java Authentication and Authorization Service** option in the **Authentication** box and select the **J2C authentication data** option;
3. Click the **New** button to open the Configuration screen;
4. Set the following fields:

Alias = dbadmin

User ID = `<database username>`

Password = `<database password>`

Description = The database security alias

where `<database username>` and `<database password>` are set to the username and password used to login to the database;

5. Press the **OK** button to confirm the changes.

Configure DB2 for z/OS Data Sources

For **z/OS** you have the choice of configuring with the Type 4 DB2 JDBC Universal Driver (XA) or the Type 2 DB2 JDBC Universal Driver (RRS).

Configuring For a Type 4 JDBC Universal Driver (XA)

Set up DB2 for z/OS Environment Variable

1. Navigate to **Environment > WebSphere variables**;
2. *Note:* The appropriate scope where the data source will be defined should be selected at this point.
3. Select the DB2UNIVERSAL_JDBC_DRIVER_PATH link from the list of environment variables. This will open the configuration screen for this variable;
4. Set the **Value** field to point to the directory containing the Type 4 drivers. This is normally the Cúram SDEJ drivers installation directory, e.g. /CuramSDEJ/drivers;
5. Press the **OK** button to confirm the changes.

Set up the Database Driver Provider

1. Navigate to **Resources > JDBC > JDBC providers**;
2. *Note:* The appropriate scope where the data source is to be defined should be selected at this point.
3. Press the **New** button to add a new driver. This will open a configuration screen;
4. Select the **DB2** drop down from the list of **database types** supplied;
5. Select the **DB2 Universal JDBC Driver Provider** drop down from the list of **Provider type** supplied;
6. Select the **XA data source** drop down from the list of **Implementation types** supplied;
7. Press the **Next** button to continue;
8. Review the properties on the configuration screen that opens. Change the Class Path line \$ {DB2UNIVERSAL_JDBC_DRIVER_PATH}/db2jcc_license_cisuz.jar to point at the DB2 for z/OS license provided by IBM for DB2 for z/OS connectivity and click **Apply**;
9. Press the **Next** and then the **Finish** button to confirm the changes.

Set up the Database Driver Data Source

The following steps should be repeated for each of the application Data Sources, substituting curamdb, curamsibdb and curamtimerdb for *<DatasourceName>* (without the angle brackets):

1. Select the DB2 Universal JDBC Driver Provider (XA) now displayed on the list of **JDBC Providers**. This will open the configuration screen for the provider;
2. Select the **Data sources** link under **Additional Properties**;
3. Press the **New** button to add a new data source;
4. Set the fields as follows:

Data source name : *<DatasourceName>*

JNDI name : *jdbc/<DatasourceName>*

Click **Next**;

5. Set the fields as follows:

Driver type : 4;

Database name : The name of the DB2 for z/OS database;

Server name : The name of the DB2 for z/OS database server;

Port number : The DB2 for z/OS database server port;

Leave all other fields untouched unless a specific change is required and click **Next**;

6. Set the fields as follows:

Set the **Component-managed authentication alias** drop down value to: *<valid for database>*;

Set the **Mapping-configuration alias** drop down value to: DefaultPrincipalMapping

Set the **Container-managed authentication alias** drop down value to: *<valid for database>*;

where the *<valid for database>* alias used is the one set up in [“Creating the Data Source Login Alias” on page 20](#);

Leave all other fields untouched unless a specific change is required and click **Next**;

7. Press the **Finish** button to confirm the changes and continue;
8. Select the newly created *DatasourceName* data source from the displayed list;
9. Select the **Custom Properties** link under **Additional Properties**;
10. Select the *fullyMaterializeLobData* entry;
11. Set the value to be *false*;
12. Click the **OK** button to confirm the change.

Configuring For a Type 2 JDBC Universal Driver (RRS)

Set up DB2 Environment Variables

1. Navigate to **Environment > WebSphere variables**;
2. *Note:* The appropriate scope where the data source will be defined should be selected at this point.
3. Select the *DB2UNIVERSAL_JDBC_DRIVER_PATH* link from the list of environment variables. This will open the configuration screen for this variable;
4. Set the **Value** field to point to the directory containing the Type 2 driver. This is normally the DB2 installation path containing the *db2jcc.jar* file.
5. Press the **OK** button to confirm the changes.
6. Select the *DB2UNIVERSAL_JDBC_DRIVER_NATIVEPATH* link from the list of environment variables. This will open the configuration screen for this variable;
7. Set the **Value** field to point to the directory containing the DB2 for z/OS shared library links for the Type 2 driver. This is the DB2 for z/OS installation path containing the Type 2 Driver libraries (such as *libdb2jcc2zos.so*, which will vary by DB2 for z/OS version and 31/64 bit implementation);
8. Press the **OK** button to confirm the changes.

Set up the Database Driver Provider

1. Navigate to **Resources > JDBC > JDBC providers**;
2. *Note:* The appropriate scope where the data source is to be defined should be selected at this point.
3. Press the **New** button to add a new driver. This will open a configuration screen;
4. Select the **DB2** drop down from the list of **database types** supplied;
5. Select the **DB2 Universal JDBC Driver Provider** drop down from the list of **provider types** supplied;
6. Select the **Connection pool data source** drop down from the list of **implementation types** supplied;
7. Press the **Next** button to continue;
8. Review the properties on the configuration screen that opens ensuring that the settings for Classpath and Native library path are correct, based on the values previously set for the environment variables *DB2UNIVERSAL_JDBC_DRIVER_PATH* and *DB2UNIVERSAL_JDBC_DRIVER_NATIVEPATH*. No changes should be required;
9. Press the **Next** and then the **Finish** button to confirm the changes.

Set up the Database Driver Data Source

The following steps should be repeated for each of the application Data Sources, substituting *curamdb*, *curamsibdb* and *curamtimerdb* for *<DatasourceName>* (without the angle brackets) in the following steps:

1. Select the DB2 Universal JDBC Driver Provider now displayed on the list of **JDBC Providers**. This will open the configuration screen for the provider;
2. Select the **Data Sources** link under **Additional Properties**;
3. Press the **New** button to add a new data source;
4. Set the fields as follows:

Data source name : *<DatasourceName>*

JNDI name : *jdbc/<DatasourceName>*
5. Click **Next** to continue;
6. Set the fields as follows:

Database name : The name of the DB2 for z/OS database;

Driver type : 2;

Leave all other fields untouched unless a specific change is required and click **Next**;
7. Set the fields as follows:

Set the **Component-managed authentication alias** drop down value to: *<valid for database>*;

Set the **Mapping-configuration alias** drop down value to: DefaultPrincipalMapping

Set the **Container-managed authentication alias** drop down value to: *<valid for database>*;

where the *<valid for database>* alias used is the one set up in [“Creating the Data Source Login Alias”](#) on page 20;

Leave all other fields untouched unless a specific change is required and click **Next**;
8. Press the **Finish** button to confirm the changes and continue;
9. Select the newly created *DatasourceName* data source from the displayed list;
10. Select the **Custom Properties** link under **Additional Properties**;
11. Select the fullyMaterializeLobData entry;
12. Set the value to be *false*;
13. Click the **OK** button to confirm the change.

Set up the JVM Property db2.jcc.propertiesFile (optional)

If you wish to use an external configuration file identified by the db2.jcc.propertiesFile property for your DB2 Type 2 Universal JDBC Driver then:

1. Navigate to **Servers > Server Types > WebSphere application servers**;
2. Select the appropriate server from the list;
3. In the **Server Infrastructure** panel expand **Java and Process Management**;
4. Select the **Process definition** link;
5. In the **processType** panel perform the following steps for each item in the list (Adjunct, Control, and Servant):
 - a. Select the **processType** link;
 - b. In the **Additional Properties** panel Select the **Java Virtual Machine** link;
 - c. In the **Additional Properties** panel Select the **Custom Properties** link;
 - d. click the **New** button and set the property as follows:

Name : db2.jcc.propertiesFile

Value : fully qualified name of the property file

Click the **OK** button to add the property.

See the information in [“WebSphere Application Server Configuration”](#) on page 8 on how to setup the property file.

Save the Master Configuration

A *Save* can be performed by clicking the **Save** link in the **Message(s)** box. This box is displayed only after configuration changes have been made.

Configure Administration Security

About this task

The default user registry used is the default WebSphere Application Server for z/OS file-based user registry.

Procedure

1. Navigate to **Security > Global security**;
2. Set the **Available realm definitions** to be **Federated repositories** and click the **Configure** button;
3. Set the **Primary administrative username** to be **websphere**;
4. Select the **Automatically generated server identity** radio button;
5. Select **Ignore case for authorization** and click the **OK** button;
6. Enter the password for the default administrative user, e.g. **websphere**, enter the confirmation and click the **OK** button to confirm the changes;
7. Select **Enable administrative security**;
8. Select **Enable application security**;
9. Select **Use Java 2 security to restrict application access to local resources** and **Warn if applications are granted custom permissions**;
10. Set the **Available realm definitions** to be **Federated repositories**
11. Click the **Apply** button to confirm the changes;
12. Navigate to **Security > Global security**;
13. Expand **Web and SIP Security** and select **Single sign-on (SSO)**;
14. Select **Requires SSL**;
15. Click **OK** to confirm the change
16. Navigate to **Security > Global Security**
17. Select the **Custom Properties** link;
18. Click the **New** button and set the name and value as follows:
Name : `com.ibm.ws.security.web.logoutOnHTTPSessionExpire`
Value : `true`
19. Click the **OK** button to add the new property.
20. Click the **New** button and set the name and value as follows:
Name : `com.ibm.ws.security.addHttpOnlyAttributeToCookies`
Value : `true`
21. Click **OK** to confirm the change
22. Save the changes to the master configuration.

Restart the Application Server

This step is compulsory. The WebSphere Application Server for z/OS address spaces must be restarted for the security changes to take effect and to add additional required users. The address spaces can be stopped using the appropriate `stopServer.sh` script in the `profiles/default/bin` directory of the

WebSphere Application Server for z/OS installation or by using the **z/OS** operator STOP command appropriate for your installation.

Before restarting the application server, it is necessary to make the registry and cryptography JAR files available to WebSphere Application Server for z/OS. The registry JAR file contains classes necessary for the security configuration and the cryptography JAR file contains necessary configuration settings and data for password security.

Registry.jar is located in the lib directory of the SDEJ installation. Copy this file into the lib directory of the WebSphere Application Server for z/OS installation.

The CryptoConfig.jar file can be generated by running the ant target configtest as follows, *build configtest -Dcrypto.ext.dir=filedir* copy the CryptoConfig.jar from the generated location. Copy this file into the Java jre/lib/ext directory. If you require customizations to the Curam cryptographic configuration see the *Curam Security Handbook* for more information.

For sites with a read-only WebSphere installation file system see the procedure [“Alternative JAR File Locations”](#) on page 10.

Now start the application server using the startServer.sh script in the profiles/default/bin directory of the WebSphere Application Server for z/OS installation or the **z/OS** operator START command appropriate for your installation and open the Administrative Console to continue with the configuration steps.

Since the security configuration is complete and the scripting changes have been made, it is now possible to use the SDEJ scripts to restart the application server. See [“Starting and Stopping WebSphere Servers”](#) on page 14 for more details on restarting the server.

The Administrative Console should now be opened to continue with the configuration. Now that global security is enabled, you will be required to login to the console with the username *websphere* and password *websphere* set up previously.

Test the DB2 for z/OS Connection

About this task

You may test your DB2 for z/OS connections once the application server has been restarted:

Procedure

1. Navigate to **Resources > JDBC > Data Sources**;
2. Check the **curamdb DataSource** and/or **curamsibdb DataSource** check box;
3. Click the **Test Connection** button;
4. The following message(s) should be displayed if successful:

Test Connection for DataSource <DataSource name> on
server <server name> at node <node name> was successful.

Otherwise, check the WebSphere Application Server for z/OS logs for details of the failure, correct, and retry.

Configure Users

About this task

As detailed in [“Security Configuration”](#) on page 10, the configured WebSphere Application Server for z/OS user registry is used for authentication of administrative users and the database user. The WebSphere Application Server for z/OS administrative users and the database user must be manually added to the user registry as follows.

Procedure

1. Navigate to **Users and Groups > Manage Users**;

2. Select the **Create** button;
3. Fill in the details for the WebSphere Application Server for z/OS administrative user and click the **Create** button.
4. Repeat the steps for the database user.

Results

Note: If WebSphere Application Server for z/OS administrative security was enabled when creating the profile the administrative user may already be defined in the registry.

Set up the System JAAS Login Module

Application security uses a JAAS (Java Authentication and Authorization Service) Login Module for authentication. This login module must be configured for the DEFAULT, WEB_INBOUND and RMI_INBOUND configurations. Repeat the below steps for each of these configurations.

Add the Login Module

1. Navigate to **Security > Global security**;
2. Expand **Java Authentication and Authorization Service** entry under the **Authentication** heading and select **System logins**;
3. Select the relevant Alias from the list. The login module should be configured for the DEFAULT, WEB_INBOUND and RMI_INBOUND aliases as follows:
4. Click the **New** button to configure a new Login Module;
5. Set the **Module class name** field to be `curam.util.security.CuramLoginModule`;
6. Check the **Use login module proxy** option;
7. Select **REQUIRED** in the **Authentication strategy** field;
8. Click the **OK** button to confirm the addition of the new login module;
9. Select the newly added `curam.util.security.CuramLoginModule` from the list;
10. Select the **Custom properties** link under the **Additional Properties** heading;
11. Click the **New** button to add the required properties as listed below.

Table 6: CuramLoginModule Custom Properties		
Name	Example Value	Description
exclude_usernames	websphere, db2admin	Required. A list of usernames to be excluded from authentication. This list should include the WebSphere Application Server for z/OS administration user (as specified in "Configure Administration Security" on page 24) and the database user (as specified in "Creating the Data Source Login Alias" on page 20). The default delimiter is a comma, but may be overridden by exclude_usernames_delimiter. Any users listed here should be defined in the WebSphere Application Server for z/OS user registry.
exclude_usernames_delimiter		Optional. A delimiter for the list of usernames provided in exclude_usernames. A delimiter other than the default comma can be useful when usernames have embedded commas as with LDAP users.
login_trace	true	Optional. This property should be set to true to debug the authentication process. If set to true the invocation of the login module will result in tracing information being added to the WebSphere Application Server for z/OS SystemOut.log file.
module_name	DEFAULT, WEB_INBOUND or RMI_INBOUND	Optional. This property should be set to one of DEFAULT, WEB_INBOUND or RMI_INBOUND depending on the configuration the login module is being defined for. It is used only when login_trace is set to true for tracing purposes.

Table 6: CuramLoginModule Custom Properties (continued)		
Name	Example Value	Description
check_identity_only	true	<i>Optional.</i> If this property is set to true the login module will not perform the usual authentication verifications. Instead it will simply ensure that the user exists on the database table. In this case the configured WebSphere Application Server for z/OS user registry will not be by-passed and will be queried after the login module. This option is intended where LDAP support is required or an alternative authentication mechanism is to be used.
user_registry_enabled	true	<i>Optional.</i> This property is used to override the behavior of by-passing the user registry. If this property is set to true the WebSphere Application Server for z/OS user registry will be queried during the authentication process. If this property is set to false, the WebSphere Application Server for z/OS user registry will not be queried. Note: If you are specifying identity only and using LDAP you may need to perform additional configuration steps; please see “Special Configuration Steps When Using Identity Only and LDAP” on page 11.
user_registry_enabled_types	EXTERNAL	<i>Optional.</i> This property is used to specify a comma-delimited list of external user types that will be processed against the WebSphere Application Server for z/OS user registry (e.g. LDAP). See “WebSphere Application Server User Registry” on page 12 for more information on the processing of the WebSphere Application Server for z/OS user registry.
user_registry_disabled_types	EXTGEN,EXTAUTO	<i>Optional.</i> This property is used to specify a comma-delimited list of external user types that will not be processed against the WebSphere Application Server for z/OS user registry (e.g. LDAP). See “WebSphere Application Server User Registry” on page 12 for more information on the processing of the WebSphere Application Server for z/OS user registry.

12. Click **OK** to confirm the addition of the new login module;

Reorder the Login Module

1. Navigate to **Security > Global security**;
2. Expand **Java Authentication and Authorization Service** under the **Authentication** heading and select **System logins**;
3. Select the relevant Alias from the list. The login module should be reordered for the DEFAULT, WEB_INBOUND and RMI_INBOUND aliases;
4. Select the **JAAS login modules** link under the **Additional Properties** heading;
5. Click the **Set Order** button;
6. Select **curam.util.security.CuramLoginModule** and click the **Move Up** button. Repeat this until the CuramLoginModule entry is the top entry in the list;
7. Click the **OK** button to confirm the modifications to the order.

Disable Cross Cluster Authentication

This property determines the behavior of a single sign-on LTPA Token2 login. The property `com.ibm.ws.security.webChallengeIfCustomSubjectNotFound` is set to `false` to ensure that web sessions can seamlessly transfer between two servers in a cluster (for example, in a fail over scenario) without being asked for security credentials.

1. Navigate to **Security > Global security**;
2. Click on **Custom properties** under the **Authentication** heading and select **com.ibm.ws.security.webChallengeIfCustomSubjectNotFound** property from the list of available properties.

3. Under General Properties, change the value of the **com.ibm.ws.security.webChallengeIfCustomSubjectNotFound** property to *false*
4. Click the **OK** button to confirm the addition;

Save the Changes

Save the changes to the master configuration as described in [“Save the Master Configuration”](#) on page 24.

Server Configuration

Configure 64-bit support

Procedure

1. Navigate to **Servers > Server Types > WebSphere application servers**;
2. Select the relevant server from the list;
3. Check the **Run in 64 bit JVM mode** check-box;
4. Click **Apply** or **OK** to apply changes;
5. Save the changes made to the master configuration using the **Save** option as before.

Results

Note: You may also need to review and adjust your JVM heap sizes based on your application size, throughput, performance goals, and other factors.

Configure your JNDI lookup port

Procedure

1. Navigate to **Servers > Server Types > WebSphere application servers**;
2. Select the relevant server from the list;
3. Expand **Ports** in the **Communications** box and press the **Details** button;
4. Select the **BOOTSTRAP_ADDRESS** entry and set the **Port** to match the value of the property `curam.server.port` in your `AppServer.properties` file;
5. Click **OK** to apply changes;
6. Save the changes made to the master configuration using the **Save** option as before.

Configure your ClassLoader settings

Procedure

1. Navigate to **Servers > Server Types > WebSphere application servers**;
2. Select the appropriate server from the list;
3. Set the **Classloader policy** to be **MULTIPLE**;
4. Click **OK** to apply changes;
5. Save the changes made to the master configuration using the **Save** option as before.

Configure your ORB Pass By Reference

Procedure

1. Navigate to **Servers > Server Types > WebSphere application servers**;
2. Select the relevant server from the list;
3. In the **Container Settings** section expand **Container Services** and click the **ORB service** link;
4. Select the **Pass by reference** option from the **General Properties** section.

5. Click **OK** to apply changes;
6. Save the changes made to the master configuration using the **Save** option as before.

Configure your Java Virtual Machine

Procedure

1. Navigate to **Servers > Server Types > WebSphere application servers**;
2. Select the appropriate server from the list;
3. In the **Server Infrastructure** panel expand **Java and Process Management**;
4. Select the **Process definition** link;
5. In the **processType** panel perform the following steps for each item in the list (Adjunct, Control, and Servant):
 - a) Select the **processType** link;
 - b) In the **Additional Properties** panel select the **Java Virtual Machine** link;
 - c) Set the fields as follows:
Initial heap size : 1024
Maximum heap size : 1024
Click **Apply** to set the values;
 - d) In the **Additional Properties** panel select the **Custom Properties** link;
 - e) Click the **New** button and set the properties as follows:
Name : com.ibm.websphere.security.util.authCacheCustomKeySupport
Value : false
Click the **OK** button to add the property;
6. Save the changes made to the master configuration using the **Save** option as before.

Configure the Timer Service

Procedure

1. Navigate to **Servers > Server Types > WebSphere application servers**;
2. Select the appropriate server from the list;
3. In the **Container Settings** panel expand **EJB Container Settings**;
4. Select the **EJB timer service settings** link;
5. In the **Scheduler Type** panel Select the **Use internal EJB timer service scheduler instance** option;
6. Set the fields as follows:
Data source JNDI name : jdbc/curamtimerdb
Data source alias : <valid for database>
where the alias used is the one set up in [“Creating the Data Source Login Alias” on page 20](#);
7. Click the **OK** button to confirm the changes;
8. Save the changes made to the master configuration using the **Save** option as before.

Set up the Port Access

Procedure

1. Navigate to **Servers > Server Types > WebSphere application servers**;
2. Select the appropriate server from the list;
3. Select the **Ports** link in the **Communications** box;

4. Select the **details** box;
5. Click the **New** button and set the following fields for the Client TCP/IP port:

User-defined Port Name : CuramClientEndPoint

Host : *

Port : *<client port>*

Set the *<client port>* to match the value of the property `curam.client.httpport` in your `AppServer.properties` file;

Click the **OK** button to apply the changes;
6. Click the **New** button and set the following fields for the WebServices TCP/IP port:

User-defined Port Name : CuramWebServicesEndPoint

Host : *

Port : *<webservices port>*

Set the *<webservices port>* to match the value of the property `curam.webservices.httpport` in your `AppServer.properties` file;

Click the **OK** button to apply the changes;
7. Navigate to **Servers > Server Types > WebSphere application Servers**;
8. Select the relevant server from the list;
9. Expand the **Web Container Settings** branch in the **Container Settings** section;
10. Select the **Web container transport chains** link;
11. Click the **New** button and set the following fields for the Client transport chain:

Name : CuramClientChain

Transport Chain Template : WebContainer-Secure

Click **Next**

Use Existing Port : CuramClientEndPoint

Click **Next** and **Finish**
12. Click the **New** button and set the following fields for the WebServices transport chain:

Name : CuramWebServicesChain

Transport Chain Template : WebContainer

Click **Next**

Use Existing Port : CuramWebServicesEndPoint

Click **Next** and **Finish**
13. Select the newly created **CuramClientChain**;
14. Select the **HTTP Inbound Channel** link;
15. Ensure the **Use persistent keep-alive connections** check-box is checked;
16. Click the **OK** button to confirm the addition;
17. Navigate to **Environment > Virtual hosts**;
18. Click the **New** button to add a new Virtual Host by setting the following fields:

Name = *client_host*

Repeat this step using the replacing *client_host* with *webservices_host*;
19. Select the **client_host** link from the list of virtual hosts;

Select the **Host Aliases** link in the **Additional Properties** box;

Click the **New** button to add a new Alias by setting the following fields;

Host Name = *

Port = <client port>

Set the <client port> to match the value of the property `curam.client.httpport` in your `AppServer.properties` file; Repeat this step for the other Virtual Host and port used (e.g. `webservices_host`)

20. Click the **OK** button to confirm the addition;
21. Save the changes to the master configuration as described in [“Save the Master Configuration”](#) on page 24.

Configure Session Security Integration

Procedure

1. Navigate to **Servers > Server Types > WebSphere application servers**;
2. Select the relevant server from the list;
3. Click **Session management** in the **Container Settings** section
4. Un-check **Security integration**. *Note: Please make sure security integration is un-checked.*
5. Click **OK** to apply changes;
6. Save the changes made to the master configuration using the **Save** option as before.

Note:

This above setting is required for IBM Cúram Social Program Management web applications.

Bus Configuration

Setup the Service Integration Bus

Procedure

1. Navigate to **Service integration > Buses**;
2. Click the **New** button and set the following field:
Name : CuramBus
Leave everything else as the default and click **Next**;
3. Entering the **Configure bus security** Wizard, Step 1.1, click **Next**;
In **Step 1.2** of the **Configure bus security** Wizard take the default setting and click **Next**;
In **Step 1.3** of the **Configure bus security** Wizard take the default setting, as appropriate, and click **Next**;
In **Step 1.4** of the **Configure bus security** Wizard review your settings and click **Next**;
4. In Step 2 click **Finish** to apply the changes.
5. Select the **CuramBus** now displayed on the list of Buses. This will open the configuration screen;
6. Select **Bus members** in the **Topology** list;
7. Click **Add** to open the **Add a New Bus Member** Wizard;
8. Select the server to add to the Bus and click the **Next** button;
9. Select **Data store** and click the **Next** button;
10. Select the option to **use existing data source** and set the options as follows:
Data source JNDI name = `jdbc/curamsibdb`
Schema name = `username`
Where `username` is the database username.
Deselect the **Create tables** option;

- Leave everything else as the default and click **Next**;
11. Take the default tuning parameters as appropriate and click **Next**;
 12. Click **Finish** to complete and exit the Wizard;
 13. Navigate to **Service integration > Buses**;
 14. Select the **CuramBus** now displayed on the list of Buses. This will open the configuration screen;
 15. Select **Security** in the **Additional Properties** section;
 16. Select **Users and groups in the bus connector role** in the **Authorization Policy** section;
 17. Click **New** to open the **SIB Security Resource Wizard**;
 18. Select the **The built in special groups** radio button and click **Next**;
 19. Select the **Server** and **AllAuthenticated** check boxes and click **Next**;
 20. Click **Finish** to complete and exit the Wizard.
 21. Save the changes to the master configuration as described in [“Save the Master Configuration” on page 24.](#)

JMS Configuration

Setup the JMS Connection Factories

Procedure

1. Navigate to **Resources > JMS > JMS providers**;
2. *Note:* The appropriate scope where the JMS resources are to be defined should be selected at this point.
3. Select the **Default messaging provider** link;
4. Select the **Connection factories** link in the **Additional Properties** box;
5. Click the **New** button and set the following fields:

Name : CuramQueueConnectionFactory

JNDI Name : jms/CuramQueueConnectionFactory

Description : The factory for all connections to the application queues.

Bus Name : CuramBus

Authentication alias for XA recovery : Same as for the jdbc/curamdb data source (e.g. <SERVERNAME> /dbadmin)

Mapping-configuration alias : DefaultPrincipalMapping

Container-managed authentication alias : Same as for the Authentication alias for XA recovery.

Leave everything else as the default and click the **OK** button to apply the changes;

6. Click the **New** button and set the following fields:

Name : CuramTopicConnectionFactory

JNDI Name : jms/CuramTopicConnectionFactory

Description : The factory for all connections to the application queues.

Bus Name : CuramBus

Authentication alias for XA recovery : Same as for the jdbc/curamdb data source (e.g. <SERVERNAME> /dbadmin)

Mapping-configuration alias : DefaultPrincipalMapping

Container-managed authentication alias : Same as for the jdbc/curamdb data source (e.g. <SERVERNAME> /dbadmin)

Leave everything else as the default and click the **OK** button to apply the changes;

7. Save the changes to the master configuration as described in [“Save the Master Configuration”](#) on page 24.

Results

Note: With the above manual configuration steps it is not possible to correctly configure security for the Curam queue and topic connection factories. To complete this part of the configuration you must use the wsadmin tool. To do so exit the Administrative Console and follow these steps:

1. Identify the queue and topic connection factory entries in the WebSphere Application Server for z/OS configuration resources.xml file. This file resides in the %WAS_HOME%\profiles \<profile_name>\config file system hierarchy depending on your naming conventions and the scope where you defined your JMS resources. For instance, using a node-level scope with a profile name of AppSrv01, a cell name of MyNodeCell and a node name of MyNode you would find this file here: C:\WebSphere\profiles\AppSrv01\config\cells\MyNodeCell\nodes\MyNode\resources.xml. In this file you must find the <factories> entities for the CuramQueueConnectionFactory and CuramTopicConnectionFactory and make note of the ID for each that begins J2CConnectionFactory_ followed by a numeric (e.g. 1264085551611).
2. Invoke the wsadmin WebSphere Application Server for z/OS script. In these examples the language is JACL, so the -lang jacl argument may need to be specified along with login credentials, etc. depending on your local configuration.
3. In wsadmin invoke the following commands; again, assuming node-scope definitions, a cell name of MyNodeCell, and a node name of MyNode, the resource IDs will be different in your environment:

- a. Get the node and cell identifier: `$AdminConfig getid /Node:MyNode`
- b. Using the node and cell identifier from the previous step, combine it and the connection factory identifier you obtained above to display the connection factory: `$AdminTask showSIBJMSConnectionFactory CuramQueueConnectionFactory(cells/MyNodeCell/nodes/MyNode|resources.xml#J2CConnectionFactory_1264085551611)`

From the above command output you should verify that authDataAlias is not set (e.g. authDataAlias=), else you're done, as shown in this sample wsadmin output:

```
{password=, logMissingTransactionContext=false,
readAhead=Default, providerEndpoints=,
shareDurableSubscriptions=InCluster,
targetTransportChain=, authDataAlias=, userName=,
targetSignificance=Preferred,
shareDataSourceWithCMP=false,
nonPersistentMapping=ExpressNonPersistent,
persistentMapping=ReliablePersistent, clientID=,
jndiName=jms/CuramQueueConnectionFactory,
manageCachedHandles=false,
consumerDoesNotModifyPayloadAfterGet=false,
category=, targetType=BusMember, busName=CuramBus,
description=None,
xaRecoveryAuthAlias=crouch/databaseAlias,
temporaryTopicNamePrefix=, remoteProtocol=,
producerDoesNotModifyPayloadAfterSet=false,
connectionProximity=Bus, target=,
temporaryQueueNamePrefix=,
name=CuramQueueConnectionFactory}
```

- c. To set the authDataAlias use the same connection factory information as above; e.g.: `$AdminTask modifySIBJMSConnectionFactory CuramQueueConnectionFactory(cells/MyNodeCell/nodes/MyNode|resources.xml#J2CConnectionFactory_1264085551611) {-authDataAlias crouch/databaseAlias}`
- d. Save the changes: `$AdminConfig save`
- e. You can invoke the showSIBJMSConnectionFactory command to verify the change.
- f. Repeat the steps above for the CuramTopicConnectionFactory.
- g. Exit the wsadmin session using the quit command, ensuring you have saved your changes.

Setup the Required Queues

About this task

Via the Administrative Console perform the following steps, substituting `<QueueName>` (without the angle brackets) with each of the following queue names: `DPEnactment`, `DPErrors`, `CuramDeadMessageQueue`, `WorkflowActivity`, `WorkflowEnactment` and `WorkflowError`.

Procedure

1. Navigate to **Service integration > Buses > CuramBus**;
2. Select the **Destinations** link in the **Destination resources** box;
3. Click the **New** button to open the "Create new destination" wizard;
4. Select **Queue** as the destination type and click **Next**;
5. Set the following queue attributes:
Identifier : `SIB_ <QueueName>`
Leave everything else as the default and click the **Next** button;
6. Use the **Selected Bus Member** and click **Next**;
7. Click **Finish** to confirm the queue creation;
8. Select the newly added `SIB_ <QueueName>` queue now displayed on the list of existing providers.
This will open the configuration screen again;
9. Use the following table to set the Exception Destination via the **Specify** radio button and associated text field;

Table 7: Exception Destination Settings	
Queue Name	Exception Destination
<code>SIB_CuramDeadMessageQueue</code>	System
<code>SIB_DPEnactment</code>	<code>SIB_DPErrors</code>
<code>SIB_DPErrors</code>	<code>SIB_CuramDeadMessageQueue</code>
<code>SIB_WorkflowActivity</code>	<code>SIB_WorkflowError</code>
<code>SIB_WorkflowEnactment</code>	<code>SIB_WorkflowError</code>
<code>SIB_WorkflowError</code>	<code>SIB_CuramDeadMessageQueue</code>

10. Click the **OK** button to apply the changes.
11. Navigate to **Resources > JMS > JMS providers**;
12. Select the **Default messaging provider** link;
13. Select the **Queues** link in the **Additional Properties** box;
14. Click the **New** button and set the following fields:

Name : `<QueueName>`

JNDI Name : `jms/ <QueueName>`

Bus Name : `CuramBus`

Queue Name : `SIB_ <QueueName>`

Delivery Mode : `Persistent`

Leave everything else as the default and click the **OK** button to apply the changes;

Results

Save the changes to the master configuration as described in [“Save the Master Configuration” on page 24.](#)

Setup the Required Topics

Procedure

1. Navigate to **Resources > JMS > JMS providers**;
2. Select the **Default messaging provider** link;
3. Select the **Topics** link in the **Additional Properties** box;
4. Click the **New** button and set the following fields:

Name : CuramCacheInvalidationTopic

JNDI Name : jms/CuramCacheInvalidationTopic

Description : Cache Invalidation Topic

Bus name : CuramBus

Topic space : Default.Topic.Space

JMS Delivery Mode : Nonpersistent

Leave everything else as the default and click the **OK** button to apply the changes;

5. Save the changes to the master configuration as described in [“Save the Master Configuration” on page 24.](#)

Setup the Required Queue Activation Specifications

About this task

As with the setting up of queues, perform these steps, substituting `<QueueName>` (without the angle brackets) with each of the following queue names: DPEnactment, DPError, CuramDeadMessageQueue, WorkflowActivity, WorkflowEnactment and WorkflowError.

Procedure

1. Navigate to **Resources > JMS > JMS providers**;
2. Select the **Default messaging provider** link;
3. Select the **Activation specifications** link in the **Additional Properties** box;
4. Create a new specification by clicking the **New** button and set the following fields:

Name : `<QueueName>`

JNDI name : eis/ `<QueueName>` AS

Destination Type : Queue

Destination JNDI name : jms/ `<QueueName>`

Bus Name : CuramBus

Authentication Alias : Same as for the jdbc/curamdb data source (e.g. `<SERVERNAME>` /dbadmin)

Leave everything else as the default and click **OK** to add the port;

Results

Save the changes to the master configuration as described in [“Save the Master Configuration” on page 24.](#)

Setup the Required Topic Activation Specifications

Procedure

1. As with the Queue Activation Specifications in the previous section, add a new Activation Specification and set the following fields:

Name : CuramCacheInvalidationTopic

JNDI name : eis/CuramCacheInvalidationTopicAS

Destination Type : Topic

Destination JNDI name : jms/CuramCacheInvalidationTopic

Bus Name : CuramBus

Authentication Alias : Same as for the jdbc/curamdb data source (e.g. <SERVERNAME> /dbadmin)

2. Leave everything else as the default and click the **OK** button to apply the changes.
3. Save the changes to the master configuration as described in [“Save the Master Configuration”](#) on page 24.

Post Configuration

Service Integration Bus Database Tables

After setup, it is necessary to manually create database tables required for the Service Integration Bus. WebSphere Application Server for z/OS provides a utility to generate the SQL for creating these tables, the SIB DDL Generator.

The generator can be run by executing the following command:

```
$WAS_HOME/bin/sibDDLGenerator.sh
-system system
-platform platform
-schema username
-database database_name
-user username
-statementend ';'
-create
```

Where

- *system* is the database that is to be used, e.g. db2;
- *platform* is the operating system, e.g. zos;
- *username* is the username required for accessing the database, as specified in the Bootstrap.properties property curam.db.username;
- *database_name* is the name of the database to be used, as specified in the Bootstrap.properties property curam.db.zos.dbname.

For example:

```
$WAS_HOME/bin/sibDDLGenerator.sh
-system db2 -platform zos
-schema db2admin -database curam -user db2admin
-statementend ';' -create
```

This command will output SQL statements to define the Service Integration Bus tables and these SQL statements must be executed on the target database.

Note: There are DB2 for z/OS -specific defaults for the STOGROUP and BUFFERPOOL; see the [WebSphere Application Server product documentation](#) for more information.

Timer Service Database Tables

After setup, it is necessary to manually create the database tables required for the Timer Service. WebSphere Application Server for z/OS provides the DDL for these tables in its \$WAS_HOME/Scheduler directory.

The DDL files that should be run are the createTablespaceDB2ZOS.ddl and createSchemaDB2ZOS.ddl, in that order.

Each DDL file contains instructions appropriate for running against your target database.

Completion

The WebSphere Application Server for z/OS is now configured and ready for installing IBM Cúram Social Program Management application .ear files. Log out of the Administration Console and restart WebSphere Application Server for z/OS using the targets description in [“Starting and Stopping WebSphere Servers”](#) on page 14.

Manual Application Deployment

About this task

To install an enterprise application in WebSphere Application Server for z/OS, the Administration Console can be used. The steps below describe how to install an application, EJB component, or web module using the Administrative Console.

Note: Once the install has been started, the **Cancel** button must be used to exit if the installation of the application is aborted. It is not sufficient to simply move to another Administrative Console page without first clicking **Cancel** on an application installation page.

Procedure

1. Navigate to **Applications > New Application**;
2. Select **New Enterprise Application**;
3. Click the appropriate radio button and specify the full path name of the source application file or .ear file, optionally via the **Browse** button, in the **Path to the new application** panel and click **Next**;
The default locations for the application .ear files is:
\$SERVER_DIR/ear/WAS/
4. Select the **Fast Path - Prompt only when additional information is required** radio button in the **How do you want to install the application?** panel and click **Next**;
5. Leave the defaults as they are for step 1, *Select installation options* and click **Next**;
6. In step 2, **Map modules to servers**, for every module listed, select a target server or a cluster from the **Clusters and Servers** list. To do this, tick the check box beside the particular module(s) and then select the server or cluster and click **Apply**.
7. For the following step(s) click **Next** and on the Summary panel click **Finish** to complete the installation. This step may take a few minutes and should finish with the message *Application Curam installed successfully*.
8. Save the changes to the Master Configuration. (See [“Save the Master Configuration”](#) on page 24 for more details.)
9. Navigate to **Applications > Application Types > WebSphere enterprise applications** and select the newly installed application.
10. Select the **Class loading and update detection** option from the **Detail Properties** section.
11. Set the **Class loader order** to be **Classes loaded with local class loader first (parent last)**.
12. Set the **WAR class loader policy** to be **Single class loader for application**.
13. Click **OK**.

14. Navigate to **Users and Groups -> Manage Users**. Click **Create...** and enter a User ID, Password, First Name and Last Name. Then click on **Create**.

See “Change SYSTEM Username” on page 17 for information regarding the credentials expected here by the application and changing them.

15. Return back to the enterprise application (**Applications > Application Types > WebSphere enterprise applications**, select the newly installed application) and select the **Security role to user/group mapping** option from the **Detail Properties** section and map the mdbuser role to a username and password as per these steps:

Note:

The username you use to map to the mdbuser role must already be defined in your user registry.

- a) Check **Select** for the mdbuser role and click **Map Users...**;
 - b) Enter an appropriate username in the **Search String** field and click **Search**;
 - c) Select the ID from the **Available:** list and click **>>** to add it to the **Selected:** list and click **OK**.
 - d) Click **OK**.
16. Having mapped the mdbuser role you can now update the user RunAs role by selecting the **User RunAs roles** option from the **Detail Properties** section.
 17. Enter the appropriate username and password in the **username** and **password** fields, respectively. Check **Select** for the mdbuser role and click **Apply**.
 18. Click **OK**.
 19. Save the changes to the master configuration.
 20. After deployment it is necessary to start the application before it can be used. Navigate to **Applications > Application Types > WebSphere enterprise applications**, tick the check box for the newly installed application, and click the **Start** button. This step may take a few minutes and should finish with the application status changing to indicate it has been started.
 21. Finally, test the application deployment. For example, point a web browser at the URL for the deployed application e.g.:

`https://<Your WebSphere host>:<CuramClientEndPoint>/Curam`

Where:

`<Your WebSphere host>` identifies the host name or IP address where your WebSphere Application Server for z/OS system is running and `<CuramClientEndPoint>` identifies the port assigned (as in “Set up the Port Access” on page 29).

WebSphere Network Deployment

IBM's WebSphere Application Server Network Deployment offers advanced deployment services, including clustering, edge services and high availability for distributed configurations.

Tips for working with WebSphere Network Deployment

Customizing for WebSphere Network Deployment

The customizing of WebSphere Network Deployment (using z/OS Profile Management Tool or ISPF) is outside the scope of this document, but along with the information you'll find in the *Program Directory for WebSphere Application Server for z/OS V7.0 (GI11-4295)* and in the Installing your application serving environment topics in the *WebSphere Application Server, Version V7.0 product documentation*. IBM offers a number of helpful Redbooks for doing this. These may be found at the IBM Redbook website: <http://www.redbooks.ibm.com/>.

Synchronizing Changes

If you are operating in a Network Deployment environment it is strongly recommended that you ensure WebSphere Application Server for z/OS synchronizes its configuration after *each* Administration Console change or Ant target.

When saving the master configuration ensure you manually force synchronization via the Administration Console:

1. Navigate to **System Administration > Save Changes to Master Repository**;
2. Check the **Synchronize changes with Nodes** check box;
3. Click the **Save** button. The synchronization may take some time;
4. Check the system and/or WebSphere Application Server for z/OS logs for synchronization completion. These messages may vary by WebSphere Application Server for z/OS release, but you are looking for something like:

ADMS0208I: The configuration synchronization complete for cell.

Once synchronization is complete, review the server status and various WebSphere Application Server for z/OS logs to ensure success;

Configuration of Node

Before deploying an application the server must first be configured. This is done through the Deployment Manager Administration Console and the configuration is then synchronized with the node's federated servers.

The Node Agent, which enables communication between the Deployment Manager and its federated servers, is required to be started. This can be done via the **z/OS** operator **START** command appropriate for your installation or the `startNode.sh` command in the `profiles/<federated profile name>/bin` directory of the WebSphere Application Server for z/OS installation.

After the Node Agent is started, all control is handed over to the Deployment Manager for this Node's servers. To start or stop a server in the Deployment Manager Administration Console:

1. Navigate to **Servers > Server Types > WebSphere application servers**;
2. Check the server to be started/stopped from the list and click the **Start** or **Stop** button as required.

The next step in the process is to configure the federated servers. As mentioned before, all configuration is done through the Deployment Manager Administrative Console. [“Manual WebSphere Application Server Configuration” on page 19](#) describes the manual WebSphere Application Server for z/OS configuration for a basic installation, and should be followed with the differences identified below. When saving the master configuration, ensure you synchronize your changes as described in [“Synchronizing Changes” on page 38](#).

[“Set up the System JAAS Login Module” on page 26](#) details the security setup required during manual configuration. This setup requires the `Registry.jar` to be copied to a directory within the WebSphere Application Server for z/OS installation. The `Registry.jar` should be copied from `CuramSDEJ/lib` to the `lib` directory of the Deployment Manager installation and any federated installations.

[“Set up the System JAAS Login Module” on page 26](#) this security setup also requires the `CryptoConfig.jar` to be copied to the `java64/lib/ext` directory within the WebSphere Application Server installation. The `CryptoConfig.jar` should be copied to the same directory structure for any other WebSphere Application Server installations in the environment.

Note: Before building the application `.ear` for deployment it is worth noting the `BOOTSTRAP_ADDRESS` of the server that these will be installed onto. The `BOOTSTRAP_ADDRESS` is located in the same list of ports as the `SOAP_CONNECTOR_ADDRESS` described previously.

By default the `BOOTSTRAP_ADDRESS` expected by the application is 2809. To solve this issue either change this address or alternatively change the relevant property in your `AppServer.properties` file.

The property that should be changed is the `curam.server.port` value in the `AppServer.properties` file. Changing this affects the port value in the `web.xml` file when building an `.ear` (EAR) file. For more information on the `web.xml` file consult the *Cúram Web Client Reference Manual*.

Deploying on the Node

Finally, [“Manual Application Deployment” on page 37](#) should be followed to manually deploy the applications on the required server. Applications can then be started or stopped using the Deployment Manager Administration Console.

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.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

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: