Cúram 8.1.2

Deploying on IBM® WebSphere® Application Server on z/OS® Guide

Note

Before using this information and the product it supports, read the information in Notices on page 51

Edition

This edition applies to Cúram 8.1, 8.1.1, and 8.1.2.

Contents

1 Deploying on IBM® WebSphere® Application Server for z/OS®

To deploy Cúram on IBM® WebSphere Application Server for z/OS®, you must first build the application EAR files on a supported build platform. On IBM® z/OS®, install and configure any needed software, and deploy the application EAR files.

1.1 Third-party tools

To use the Cúram application, you must install and configure third-party software.

It is beyond the scope of this topic to give detailed information of all IBM® z/OS® specific software and their various dependencies that are required to support Cúram.

The prerequisites, installation notes, or postinstallation configuration activities that are relevant to Cúram are outlined for the following:

IBM® Db2® for z/OS®

Install IBM® Db2® for z/OS® and perform Cúram-specific postinstallation tasks.

Prerequisites and Installation

See the program directory for IBM® Db2® for z/OS® as specified in Db2 12 for z/OS Product Documentation that corresponds to a Cúram-supported version. See also Cúram Supported Prerequisites.

Install IBM® Db2® for z/OS® as described in the relevant IBM® Db2® for z/OS® product documentation.

Postinstallation

Create a target Db2® subsystem that contains the database as outlined in the Db2® 12 product documentation, see Db2 12 for z/OS Product Documentation.

There are many subsystem configuration options, note the following options for Cúram

- Because Cúram and application interfaces require JDBC type 4 connectivity, DDF must be configured for the subsystem.
- The RRULOCK DSNZPARM parameter must be set to YES.
- Setting the IDTHTOIN DSNZPARM parameter too low might cause Cúram batch errors such as those in the following shell output:

```
[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.
```

  or a DSNL027I timeout message with a reason code of 00D3003B in the IBM® Db2® for z/OS® or system log.

- Setting the LOBVALA and LOBVALS DSNZPARM parameters too low can cause IBM® Db2® for z/OS® `resource unavailable` errors. A specific value depends on available local resources, performance targets, but, values no lower than the following are a suggested starting point:

```
LOBVALA=192000
LOBVALS=32768
```

When the database is installed and a target Db2® subsystem is configured on IBM® z/OS®, you must create a Cúram database. As with creating a Db2® subsystem there are many options available depending on your site requirements such as security and physical database design.

For Cúram, starting SQL by using Ant scripts requires that the Db2® database storage group function is available. For example, before you create the database, create a storage group (where the values in angle brackets are your local names and values:

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

Then, when you create a Cúram database you must reference the storage group with the STOGROUP option on the **`CREATE DATABASE SQL`** statement.

A Cúram database must be created before you run the Ant **`database`** and **`prepare.application.data`** targets as described in *Building a database*.

For example:

```
CREATE DATABASE CURAM BUFFERPOOL BP0 INDEXBP BP0 STOGROUP <storage_group> ;
```

For any shell that accesses a C ram IBM® Db2® for z/OS® database, the environment variable *DB2JCC_LICENSE_CISUZ_JAR*, must be defined that points to the fully qualified path of the *`db2jcc_license_cisuz.jar`* file.

Remote IBM® Db2® for z/OS® connectivity

Before a connection can be established to a remote database, it must be configured. The full details of installation of IBM® Db2® for z/OS® are beyond the scope of this document set. However, note the following the main postinstallation steps.

- A database can be configured for EBCDIC, ASCII, or UNICODE mode for use by the application. You can do this when creating the database by using theCCSID keyword. For ASCII or UNICODE databases, the `curam.db.zos.encoding` property is required, see .

  `CREATE DATABASE <database_name> BUFFERPOOL BP0 INDEXBP BP0 STOGROUP <storage_group> CCSID <EBCDIC, ASCII or UNICODE>;`
- An environment variable called `DB2JCC_LICENSE_CISUZ_JAR` must be created that points to the installed IBM® Db2® for z/OS® license jar file used for connectivity to the remote database server. This is normally named *`db2jcc_license_cisuz.jar`* and is provided with IBM® Db2® for z/OS® or Db2® Connect.

IBM® WebSphere® Application Server for z/OS®

Install WebSphere® Application Server for z/OS®, and perform postinstallation tasks.

Prerequisites and installation

See Cúram supported prerequisites for the supported versions of WebSphere® Application Server for z/OS®. Then, follow the WebSphere Application Server for z/OS documentation to install it.

Postinstallation

As part of installing WebSphere® Application Server for z/OS® you create a stand-alone application server cell. This cell is where you configure the application server as described in 1.5 Application server configuration on page 17.

Based on the name of the cell and the file system you create as part of installing the application server, you must define Ant targets in the *WAS_HOME*environment variable. *WAS_HOME* must point to the *`WebSphereAppServer`* folder; for example:

```
export WAS_HOME=/IBM/WebSphere/X1CELL/AppServer
```

Security

Generally, Cúram application security is contained within the Java® EE and application server contexts. Integrating WebSphere® Application Server for z/OS® and Cúram security with RACF is outside the scope of Cúram as delivered.

Apache Ant

Install Apache Ant and perform postinstallation tasks.

Apache Ant is a Java™ build tool. Apache Ant is similar to the *make* tool.

Installation

1. Obtain a Cúram-supported version of Ant from the Apache site. For more information, see Cúram Supported Prerequisites
2. Extract the .zip file to the target file system; for example:

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

   Where `<version>` represents the relevant Ant version.

Postinstallation

1. Ensure the Ant script in *`apache-ant-<version>/bin`* is in EBCDIC format; for example:

   ```
   iconv -t IBM-1047 -f ISO8859-1 apache-ant-<version>/bin/ant \
   > /tmp/ant
   mv /tmp/ant apache-ant-<version>/bin
   ```

   Where `<version>` represents the relevant Ant version.

2. Make the Ant script executable; for example:

```
chmod a+x apache-ant-<version>/bin/ant
```

Where `<version>` represents the relevant Ant version.

For any shell that uses Cúram Ant targets

1. Environment variable *ANT_HOME* must be defined and must point to the Ant installation folder, for example:

```
export ANT_HOME=/usr/local/apache-ant-<version>
```

Where `<version>` represents the relevant Ant version.

2. Add Ant to the PATH; for example:

```
export PATH=$ANT_HOME/bin:$PATH
```

3. Define environment variable, ANT_OPTS as follows:

```
export ANT_OPTS='-Xmx1400m -Dcmp.maxmemory=1400m'
```

4. Test Ant by running the following command:

```
ant -version
```

You should see output that shows the Ant version and compilation date.

Java™ SE and Java™ EE

Java™ SE and Java™ EE are contained within the IBM® WebSphere® Application Server for z/OS® installation.

For any shell that uses Cúram, you must define Ant targets environment variables as follows:

- *JAVA_HOME* must point to the installed Java™. Typically, this variable would be based on *$WAS_HOME*, for example:

```
export JAVA_HOME=$WAS_HOME/java
```

- Java™ must be added to the PATH, for example:

```
export PATH=$JAVA_HOME/bin:$PATH
```

- J2EE_JAR must point to the IBM® WebSphere® Application Server for z/OS® Java™ EE JAR file, for example:

```
export $WAS_HOME/lib/j2ee.jar
```

1.2 Building EAR files

Before deploying an Cúram application the application EAR files must be built on a platform that supports building of Cúram. For more information see Cúram Supported Prerequisites.

In summary, the building of application EAR files involves invoking the Ant server, client, WebSphere® Application Server EAR, and release targets on a supported build platform. See *Building and configuring a Cúram application* for more information to perform these builds.

Related information

1.3 Installing Cúram on IBM® z/OS®

When Cúram is built, it must be installed on IBM® z/OS®

Running the Ant **`release`** target on the build platform with the **`-Dcreate.zip=true`** command line argument creates a *`release.zip`* file in the *`$SERVER_DIR/release`* folder on the build machine. Copy *`release.zip`* file to the target IBM® z/OS® file system.

Take the following steps on IBM® z/OS®:

1. Unzip the release.zip file in the target folder.
2. Convert the *`build.sh`* and *`SetEnvironment.sh`* scripts to EBCDIC format; for example:

```
iconv -t IBM-1047 -f ISO8859-1 build.sh > /tmp/build.sh
mv /tmp/build.sh .
iconv -t IBM-1047 -f ISO8859-1 build.sh > /tmp/SetEnvironment.sh
mv /tmp/SetEnvironment.sh .
```

3. Make the *`build.sh`* file executable; for example:

```
chmod u+x build.sh
```

1.4 Property files

To run Cúram Ant targets, you must have the required property files in the *`$SERVER_DIR/project/property`* directory.

The following topics identify relevant contents of these property files for IBM® z/OS® and how to verify those contents. For more information about Cúram properties, see *Application properties*.

Related information

Bootstrap properties

The *Bootstrap.properties* file contains the machine-specific configuration properties to connect to the database.

Note: *Bootstrap.properties* must be stored in ASCII format. EBCDIC format is not readable by the Ant tool.

Pay specific attention to the following elements:

Table 1: IBM® z/OS® for Db2-specific database properties

Property	Notes
curam.db.type	Value must be set to “zos”.
curam.db.zos.enableforcecheck	The default value is "false". If set to "true", the IBM® Db2® tables are put into CHECK_PENDING state and must be reset through direct DBA intervention.
curam.db.zos.encoding	Specify the value EBCDIC, ASCII, or UNICODE, that corresponds to the encoding of the database. EBCDIC is the default value.
curam.db.zos.dbname	Specify the IBM® Db2® for z/OS® database name from the CREATE DATABASE SQL statement.
curam.db.zos.32ktablespace	Specify a table space name that is created in the database that is defined to the 32 K buffer pool.
curam.db.username	Specify the z/OS user name that connects to the IBM® z/OS® database.
curam.db.password	Specify the encrypted password for the user specified by curam.db.username. Generate the encrypted value by starting the Ant encrypt target; for example: `cd $SERVER_DIR; ./build.sh encrypt -Dpassword=<The password for curam.db.username>`
curam.db.name	Specify the IBM® Db2® for z/OS® location name.
curam.db.servername	Specify the IBM® z/OS® hostname or IP address of the IBM® Db2® for z/OS® server.
curam.db.serverport	Specify the IBM® Db2® for z/OS® port.

AppServer properties

AppServer.properties contains properties that are related to the application server environment.

Tables 1 to 3 identify the most relevant properties in *AppServer.properties*. However, there are more security-related properties that are covered in more detail elsewhere. For more information, see *Default Configuration for WebSphere*.

Note: This file must be stored in ASCII format. EBCDIC format is not readable by the Ant tool.

Pay specific attention to the following properties:

- Structure-related properties for WebSphere® Application Server for z/OS® are shown in Table 1.

Table 2: Structure-related properties

Property	Value
curam.server.host	The hostname or IP address of your IBM® z/OS® host.
curam.server.name	Must match the name of the target WebSphere® Application Server for z/OS® server name that is established during its configuration.
cell.name	Must match the name of the target WebSphere® Application Server for z/OS® cell name that is established during its configuration.
node.name	Must match the name of the target WebSphere® Application Server for z/OS® node name that is established during its configuration.
profile.name	For WebSphere® Application Server for z/OS®, the profile name is always "default".
as.vendor	Value must be set to **IBM**.
curam.server.jvm.heap.size	Used by the Ant **`configure`** target to set the JVM heap size to the specified number of megabytes. The default value is 1024. Monitor the size and performance of your JVM heap to determine an optimal value.

- Port-related properties for WebSphere® Application Server for z/OS® are shown in Table 2.

Table 3: Port-related properties

Property	Value
curam.server.port	Must match the application server RMI port value.
curam.client.httpport	Must be bound to the CuramClientEndPoint port definition as described in Set up the port access.
curam.webservices.httpport	Must be bound to the CuramWebServicesEndPoint port definition as described in Set up the port access.

- *Table 4: Security-related properties*

Property	Value
security.username	The user name for the WebSphere® Application Server for z/OS® administrator.
security.password	The encrypted password for the WebSphere® Application Server for z/OS® administrator. Generate the encrypted value by starting the Ant **`encrypt`** target. For more information, see Cúram Server Developers Guide.
curam.security.credentials.async.username	The user name JMS invocations must run under, the default value is SYSTEM.
curam.security.credentials.async.password	The encrypted password associated with the JMS user name. The password must be encrypted by using the Ant **`encrypt`** target. For more information, see the Cúram Server Developers Guide.

Related information

Checking the configuration

Check that your property files are configured correctly.

Check the property files and configuration by running the Ant **configtest** target from the folder where you extracted the release.zip file that is described in 1.3 Installing Cúram on IBM® z/OS® on page 13. Run the following commands:

```
SetEnvironment.sh
./build.sh configtest
```

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

Alternative crypto JAR file locations

If the IBM® WebSphere® Application Server for z/OS® configuration file system is mounted as read-only, the placement of the Cúram registry and cryptography JAR files fail.

By default, each time that the Ant **configtest** target is run, the JAR files are copied into the configuration file system directories *$JAVA_HOME/lib/ext* and *$WAS_HOME/lib*. If the underlying installation file system is mounted as read-only, the copy of the JAR files fails. You might not be able to remount the installation file system as read/write for each invocation of the Ant **configtest** target. However, you can set up a symbolic link with the file system mounted as read/write, which is a one-time activity. You can then specify an alternative location for the files to be copied to.

The following steps show the one-time procedure:

1. Mount the installation file system, for example, */usr/lpp/zWebSphere/*, as read/write.
2. Create a symbolic link in the *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® Application Server for z/OS® installation file system as read-only.

The previous steps enable your IBM® WebSphere® Application Server for z/OS® file system to remain read-only when the Ant **configtest** target is run, which copies the files to the alternative location that points to the installation file system. When you run the Ant **configtest** 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/
```

1.5 Application server configuration

You can configure an application server either by using the Ant **configure** target, or by using the WebSphere® Application Server administrative console.

Before you configure the application server, you must install IBM® WebSphere® Application Server for IBM® z/OS®. For more information, see 1.1 Third-party tools on page 9.

Use one of the following methods to configure an application server:

- By using the Ant **configure** target. This approach applies to simple single-cell test environments.
- By using the WebSphere® Application Server administrative console as described in 1.7 Manual application server configuration on page 25. This approach applies to more complex clustered environments.

A number of Ant targets are available to help you configure and deploy Cúram as described in the following topics.

The Ant **configure** target that is provided by Cúram represents a simple default configuration and might not be suitable for a production environment.

> **Note:** On WebSphere® Application Server for IBM® z/OS® the only profile available is the default profile. No other option is possible.
>
> Therefore, before you configure the application server, either by using the Ant **configure** target or the administrative console, back up the WebSphere® Application Server configuration file system in case you need to rerun the configuration process.

The following topics provide more information about the configuration process and options:

Configuring IBM® WebSphere® Application Server for z/OS®

Use the Ant **configure** target to configure WebSphere® Application Server for z/OS®

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. Perform all these tasks by running the Ant **configure** target.

Run the Ant **configure** target from the *$SERVER_DIR* directory as follows:

```
./build.sh configure
```

This target requires that the *AppServer.properties* and *Bootstrap.properties* files are in the *$SERVER_DIR/project/properties* directory. For more information, see 1.4 Property files on page 13.

Configuring the JVM Size

By default the Ant configure target sets the application server JVM initial and maximum heap size. 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 before you run the target.

The provided heap size default is a starting point. Based on the size of your customized application, deployment strategy, etc. this size might be too low or too high. Determine the optimum value by monitoring the JVM memory performance of your server.

Configuring a Db2 type 2 Driver

By default the Ant **`configure`** target establishes a Db2® Universal type 4 Driver (XA) data source. However, 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® installation path.

There are a number of ways to configure IBM® 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 IBM® z/OS® jcc property file that is set in the servant JVM db2.jcc.propertiesFile property, which may contain various settings such as the subsystem ID.

Configuring security

Configure security settings for Cúram within WebSphere® Application Server for z/OS®

The default security configuration of Cúram involves a JAAS Login Module for Cúram users and the default WebSphere® Application Server file-based user registry for the WebSphere® Application Server and IBM® Db2® users. For more information, see the Default Configuration for WebSphere® Application Server section in Configuring security.

Alternative security configurations are available that can be used with WebSphere® Application Server for z/OS®, such as an LDAP directory server or a single sign-on solution. The following sections describe these configurations.

Related information

Configuring identity only and LDAP

Configure IBM® WebSphere® Application Server for z/OS® with the Ant **`configure`** target to use identity only and LDAP

About this task

To configure WebSphere® Application Server for z/OS® with the Ant **`configure`** target to use identity only and LDAP, set the *curam.security.check.identity.only* property to true in *`AppServer.properties`* before you run the Ant **`configure`** target. This step configures the CuramLoginModule behavior to be compatible with the identity only and LDAP authentication mechanism. For a previously configured server, see the steps in Set up the System JAAS Login Module. To set the corresponding *CuramLoginModule check_identity_only* property.

However, when you use identity only in combination with WebSphere® Application Server for z/OS® and LDAP you might need to perform extra manual configuration steps. perform these steps regardless of whether configuration is done by using the WebSphere® Application Server for z/

OS® administrative console or the Ant **configure** target. For more information, see Identity Only Authentication.

Troubleshooting Identity Only and LDAP

With this combination, you might find that WebSphere® Application Server for z/OS® fails to start successfully because you must add a WebSphere Application Server for z/OS-generated user name to the login module exclude list property (exclude_usernames). If the application server fails to start, there will be a SECJ0270E error message in the *SystemOut.log* file before the failure.

Take the following steps to resolve this error:

Procedure

1. Identify the user name that is causing application server startup to fail. Configure the login module trace as described in Logging the authentication process on page 21 (in regard to the Ant **configure** target) or Adding the login module on page 34 (in regard to configuring using the administrative console), and restart the application server. With the login module trace running, before the SECJ0270E error in the *SystemOut.log* file, the trace data identifies the failing user name with a record like this:

   ```
   SystemOut     O 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 is the error, which looks 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 user name in the login module exclude_usernames property in the application server configuration. Since the application server fails to start, you cannot make this change by using the administrative console, 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 has three occurrences of the exclude_usernames property (one for each alias); for example:

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

   You must modify the three occurrences to include the newly identified user name from the trace entry for example:

   ```
   <options xmi:id="Property_1301940482165"
      name="exclude_usernames"

    value="websphere,db2admin,server:MyNodeCell_MyNode_CuramServer"
      required="false"/>
   ```

   In the exclude_usernames occurrences the `id` attribute varies depending on your system configuration. Also, he comma separator in the example value attribute represents the default curam.security.usernames.delimiter value, which might be different in your configuration.

3. Restart WebSphere® Application Server for z/OS®.

User registry

By default, the configured IBM® WebSphere® Application Server for z/OS® user registry is not queried as part of authentication. When the *CuramLoginModule* login module is configured for identity only, the WebSphere® Application Server for z/OS® user registry is queried.

You can override this default behavior before you run the Ant **`configure`** target to set the *curam.security.user.registry.enabled* property in *AppServer.properties*. For a previously configured server, see the steps in Setting up the system JAAS login module on page 34 to set the corresponding *CuramLoginModuleuser_registry_enabled* property.

If this property is set to true the user registry is queried during the authentication process, regardless of whether identity only authentication is enabled or disabled. If this property is set to false, the user registry is not queried. For example, if *curam.security.check.identity.only* (*check_identity_only*) is set to true and *curam.security.user.registry.enabled* (*user_registry_enabled*) is set to false, neither the Cúram authentication verifications nor the WebSphere® Application Server for z/OS® user registry is used as part of the authentication process.

You can also control the authentication of different types of external users against the WebSphere® Application Server for z/OS® user registry by using the *curam.security.user.registry.enabled.types* and/or the *curam.security.user.registry.disabled.types* properties in `AppServer.properties` before you running the Ant **`configure`** target. For a previously configured server, see the steps in Setting up the system JAAS login module on page 34 to set the corresponding *CuramLoginModule user_registry_enabled_types* or *user_registry_disabled_types* properties. The following properties specify a comma-delimited list of external user types that might be authenticated by using the WebSphere® Application Server for z/OS® user registry:

- User types specified in the curam.security.user.registry.enabled.types (user_registry_enabled_types) list are processed against the user registry (for example, LDAP) and your *ExternalAccessSecurity* implementation. See Securing the application for more information on *ExternalAccessSecurity* implementations.
- User types specified in the *curam.security.user.registry.disabled.types* (*user_registry_disabled_types*) list are not processed against the user registry and the processing of your ExternalAccessSecurity implementation is the authority for authentication.

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

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

Logging the authentication process

Trace entries of the *CuramLoginModule* authentication process can be generated in `SystemOut.log` and used for debugging.

To generate these trace log entries add:

```
curam.security.login.trace=true
```

To `AppServer.properties` before you run the Ant **`configure`** target.

For a previously configured server, see the steps in Set up the System JAAS Login Module to set the *CuramLoginModule login_trace* property.

Establishing an alternate exclude user name delimiter

In rare cases, user names might conflict with the character (a comma) used to separate the list of user names that the *CuramLoginModule* excludes from authentication. In this case, an alternate character can be specified.

To specify a different character, add for example:

```
curam.security.usernames.delimiter=|
```

to `AppServer.properties` before you run the Ant **`configure`** target.

For a previously configured server, see the steps in Set up the System JAAS Login Module to set the *exclude_usernames_delimiter* property.

Application server caching behavior

IBM® WebSphere® Application Server for z/OS® caches user information and credentials in a security cache and the application login module is not launched while a user entry is valid in this cache.

IBM® WebSphere® Application Server for z/OS® caches user information and credentials in a security cache and the application login module is not launched while a user entry is valid in this cache. The default invalidation time for this security cache is ten minutes, where the user is inactive for ten minutes. For more information, see WebSphere caching behavior.

Related information

Security custom properties

Use the *com.ibm.ws.security.webChallengeIfCustomSubjectNotFound* to determine the behavior of a single sign-on LTPA Token2 login.

com.ibm.ws.security.webChallengeIfCustomSubjectNotFound

The *com.ibm.ws.security.webChallengeIfCustomSubjectNotFound* 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. Then the token is used to log in directly because the custom information needs to be gathered again. Therefore, a challenge occurs requiring the user to log in 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 Cúram configuration script sets the 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 failover scenario) without being asked for security credentials. This setting allows the security token that is used by WebSphere® Application Server for z/OS® to be validated correctly, without user input.

If this behavior is not required, you can change this property to true, see Setting up the system JAAS login module on page 34 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

Set the security **Requires SSL** option and set a custom property to limit LTPA cookies to SSL only.

When users log in to the application, their usernames and passwords are sent to the server, and if successfully authenticated, the server responds with a unique LTPA token. This token is used in all subsequent requests to recognize the user and then serves privileged content. When the user logs out, this token doesn't become invalid. 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:

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

The default configuration scripts make these changes and the steps are documented Configuring administration security on page 31.

Cúram cryptography

Cúram cryptography relates to functions for managing passwords.

Cúram cryptography is covered in detail in Configuring Security. Consult this reference with the following in mind:

- For production environments, you must modify the default settings.
- For development and test environments, consider whether the defaults provide acceptable protection in your environment.

Starting and stopping WebSphere® Application Server for z/OS®

Use the Ant **`startserver`** and **`stopserver`** targets to start and stop WebSphere® Application Server for z/OS® servers.

The Ant targets require the following cconfiguration steps:

- Run the target from the *`$SERVER_DIR`* directory.
- Set up the *`AppServer.properties`* file as described in 1.4 Property files on page 13.
- Specify the *server.name* property that specifies the name of the application server on the command line by using **`-Dserver.name=`**. Alternatively, specify *server.name*

directly in *AppServer.properties*. The *server.name* property can be added to the *AppServer.properties* file along with *curam.server.name*.

Example commands are as follows:

- To start a server:

```
cd $SERVER_DIR
. SetEnvironment.sh
./build.sh startserver -Dserver.name=CuramServer
```

- To stop a server:

```
cd $SERVER_DIR
. SetEnvironment.sh
./build.sh stopserver -Dserver.name=CuramServer
```

- To restart a server:

```
cd $SERVER_DIR
. SetEnvironment.sh
./build.sh restartserver -Dserver.name=CuramServer
```

Running **restartserver** when the server is already stopped starts it.

Related concepts

Configuring IBM WebSphere Application Server for z/OS on page 17
Use the Ant **configure** target to configure WebSphere® Application Server for z/OS®

1.6 Deployment

To deploy an application, you install it. To install an updated version of an application, you must first uninstall it.

Precompiling JSPs

To improve initial page load performance, use the Ant **precompile** target to precompile the JSPs in an application EAR file before deployment.

The Ant **precompile** target requires the following argument on the command line:

```
 -Dear.file=
```

The fully qualified path of the EAR file to install.

An example of invocation is as follows:

```
./build.sh precompilejsp -Dear.file=$SERVER_DIR/build/ear/WAS/Curam.ear
```

Note: Depending on factors such as the capabilities of your system and the size of the EAR file, the **precompile** target can take up to several hours to complete. Ensure that your environment is not significantly restricted regarding available CPU, memory, and free space in the *$CURAMSDEJ* file system.

Deploying an application

Before you deploy or install a Cúram application, you must change the JMS credentials.

You must undeploy or uninstall an application before you install an updated version of the application.

Install an application

Use the Ant **`installapp`** target to install the Cúram EAR files.

The Ant target requires the following arguments:

- -Dear.file The fully qualified name of the EAR file to install.
- -Dapplication.name The name of the application.
- -Dserver.name The name of the server to install the application, which can alternatively be specified in *`AppServer.properties`*.

Example of Usage:

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

Note: The EAR file that contains the server module must be deployed before you install any other client-only EAR files.

Use the following optional Ant property to pass more arguments to the IBM® WebSphere® Application Server for z/OS® **`wsadmin`** command:

```
wsadmin.extra.args
```

For example, the following command 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 might need to escape the quotation marks, for example:

```
-Dwsadmin.extra.args=\"-appendtrace true\"
```

Uninstall an application

If you need to install a new version of an application, use the Ant **`uninstallapp`** target to uninstall the application.

Ant **`uninstallapp`** target requires the following arguments to be specified on the command line:

- -Dapplication.name=The name of the application to uninstall (as configured during installation).
- -Dserver.name=The name of the server the application is installed on. The name can alternatively be specified in *`AppServer.properties`*.

A usage example is as follows:

```
./build.sh uninstallApp -Dapplication.name=Curam -Dserver.name=CuramServer
```

Populating the database

Before you access Cúram, you must populate the database with default data.

Use the Ant **`database`** and **`prepare.application.data`** targets to populate the database:

```
./build.sh database prepare.application.data
```

Testing the deployment by logging in to the application

When the application is deployed, log in to the application to display the application landing page, this action verifies basic functions of the application.

Ensure that the relevant server is started and enter the application URL in a web browser, for example:

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

Where the components of the URL are as follows:

- *<some.machine.com>* identifies the hostname or IP address where WebSphere® Application Server for z/OS® is running.
- *<port>* identifies the server port on which WebSphere® Application Server for z/OS® is listening. For more information, see `curam.client.httpport` in 1.4 Property files on page 13.
- *<context-root>* identifies the context root of the WAR module, which would typically be **Curam**.

For the Cúram, the URL directs you to the application login page. Log in with a valid user name and password, the browser is then redirected to the application landing page.

Some applications, like the CitizenPortal context root, don't require an explicit login and the landing page is entered directly. For more information about CitizenPortal, see *Merative ™ Cúram Universal Access*.

1.7 Manual application server configuration

You can configure an application server installation manually, if required. Use the following manual steps to configure and deploy a base installation of IBM® WebSphere® Application Server for z/OS®.

Do not configure an application server manually when you are using a base application server installation. Instead, use the Ant **configure** target. For the equivalent steps that are done by the Ant **configure** target, including necessary background information and infrastructure changes, see Configuring IBM® WebSphere® Application Server for z/OS® on page 17.

To deploy an application server manually in a network deployment installation, you must adapt the steps. For more information, see Network deployment on page 48.

The administrative console

Configure IBM® WebSphere® Application Server for z/OS® by using the administrative console.

To run the administrative console, you must start the server in the default profile. This step is required because the console is installed as a web application on this server. For more information, see Starting and stopping WebSphere® Application Server for z/OS® on page 22.

To open the console, point a web browser at the following URL:

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

Where:

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

Any settings that are 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. That is, if you are working in a cluster it might not be necessary to set the same settings on each server, so the scope may be set to cell or node.

Scripting support

To support the execution of provided Ant scripts, change the IBM® 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 IBM® WebSphere® Application Server for z/OS® installation. It is necessary to set the login source to retrieve the user name 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 user name and password for the administrative console.

soap.client.props

Open the *soap.client.props* file, also found in the *profiles/default/properties* directory the IBM® WebSphere® Application Server for z/OS® installation. It is necessary to set the login source to retrieve the user name 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 Configuring IBM® WebSphere®

Application Server for z/OS® on page 17. In the example below the values are merely examples and the password that is specified in this file cannot be encrypted:

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

where *websphere* is the user name and password for the administrative console .

To avoid timeouts when you instal application EAR files, ensure that the following is set, for example:

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

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

server.policy

Open the *server.policy* file found in the *profiles/default/properties* directory of the IBM® 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 file.

Creating the data source login alias

Use the administrator console to create the data source login alias.

About this task

IBM® Db2® for z/OS® is the database that is supported on IBM® z/OS®. The IBM® WebSphere® Application Server for z/OS® administrative console is used to configure a login alias for the IBM® Db2® for z/OS® data sources as follows:

Procedure

1. Browse 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 **New** 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 the user name and password that is used to log in to the database.

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

Configure IBM® Db2® for IBM® z/OS® data sources

Configure IBM® Db2®. You can either configure by using 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)

Configure for a type 4 JDBC universal driver (XA) by setting up the IBM® Db2® for z/OS® environment variable, the database driver provider, and the database driver data source.

Set up IBM® Db2® for z/OS® environment variable

1. Browse to **Environment** > **WebSphere variables**. The appropriate scope where the data source is defined is selected.
2. Select the **DB2UNIVERSAL_JDBC_DRIVER_PATH** link from the list of environment variables.
3. Set the **Value** field to point to the directory that contains the Type 4 drivers. This directory is normally the SDEJ `drivers` installation directory, for example: `/CuramSDEJ/drivers`.
4. Press **OK** to confirm the changes.

Set up the database driver provider

1. Browse to **Resources** > **JDBC** > **JDBC providers**. The appropriate scope where the data source is to be defined is selected.
2. Press **New** to add a driver.
3. Select **DB2** from the list of **database types** supplied.
4. Select **DB2 Universal JDBC Driver Provider** from the list of **Provider type** supplied.
5. Select **XA data source** from the list of **Implementation types** supplied.
6. Press **Next** to continue.
7. 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 license that is provided for IBM® Db2® for z/OS® connectivity and click **Apply**.
8. Press **Next** > **Finish** to confirm the changes.

Set up the database driver data source

Repeat the following steps 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**.
2. Select the **Data sources** link under **Additional Properties**;

3. Press **New** to add a 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 database
   - **Server name**: The name of the database server
   - **Port number**: The database server port

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

   - Set **Component-managed authentication alias** to: *<valid for database>*
   - Set **Mapping-configuration alias** to: DefaultPrinicipalMapping
   - Set **Container-managed authentication alias** to: *<valid for database>*

   Where the *<valid for database>* alias is the one that you set up in Creating the data source login alias on page 27.

   Leave all other fields unchanged unless a specific change is required and click **Next**.
7. Select **Finish** 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 **OK** to confirm the changes.

Save the master configuration

Click the **Save** link in the **Message(s)** box to save the master configuration. This box is displayed only after configuration changes are made.

Configuring for a type 2 JDBC universal driver (RRS)

Configure a type 2 JDBC universal driver (RRS) by setting up the IBM® Db2® for z/OS® environment variable, the database driver provider, and the database driver data source. Optionally, you can also set up the JVM property Db2.jcc.propertiesFile.

Set up IBM® Db2® environment variables

1. Browse to **Environment** > **WebSphere variables**. The appropriate scope where the data source is defined is selected.
2. Select the **DB2UNIVERSAL_JDBC_DRIVER_PATH** link from the list of environment variables.
3. Set the **Value** field to point to the directory that contains the Type 2 driver. This directory is normally the Db2® installation path that contains the *`db2jcc.jar`* file.
4. Press **OK** to confirm the changes.

5. Select the DB2UNIVERSAL_JDBC_DRIVER_NATIVEPATH link from the list of environment variables.
6. Set the **Value** field to point to the directory that contains the IBM® Db2® for z/OS® shared library links for the Type 2 driver. This directory is the installation path that contains the Type 2 Driver libraries (such as `libdb2jcct2zos.so`, which varies by IBM® Db2® for z/OS® version and 31 or 64-bit implementation)
7. Press **OK** to confirm the changes.

Set up the database driver provider

1. Browse to **ResourcesJDBCJDBC providers**. The appropriate scope where the data source is to be defined is selected.
2. Select **New** to add a driver.
3. Select **DB2** from the list of **database types** supplied
4. Select **DB2 Universal JDBC Driver Provider** from the list of **provider types** supplied
5. Select **Connection pool data source** from the list of **implementation types** supplied
6. Press **Next** to continue
7. Review the properties on the configuration screen that opens ensuring that the settings for `Classpath` and `Native library` paths are correct. Review the properties based on the values that you set for the environment variables DB2UNIVERSAL_JDBC_DRIVER_PATH and DB2UNIVERSAL_JDBC_DRIVER_NATIVEPATH. No changes should be required
8. Press **Next** > **Finish** to confirm the changes.

Set up the database driver data source

Repeat the following steps for each of the application data sources, substituting curamdb, curamsibdb , and curamtimerdb for <*DatasourceName*> (without the angle brackets) in the following steps:

1. Select DB2 Universal JDBC Driver Provider from the list of **JDBC Providers**.
2. Select the **Data Sources** link under **Additional Properties**
3. Press **New** to add a data source
4. Set the fields as follows:
   - **Data source name**: <*DatasourceName*>
   - **JNDI name**: *jdbc/*<*DatasourceName*>
5. Select **Next** to continue
6. Set the fields as follows:
   - **Database name**: The name of the IBM® Db2® for z/OS® database
   - **Driver type**: 2
   - Leave all other fields untouched unless a specific change is required and select **Next**
7. Set the fields as follows:
   - Set **Component-managed authentication alias** to: <*valid for database*>
   - Set **Mapping-configuration alias** to: DefaultPrinicipalMapping
   - Set **Container-managed authentication alias** to: <*valid for database*>

   Where the <*valid for database*> alias is the one that you set up in Creating the data source login alias on page 27

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

8. Select **Finish** to confirm the changes.
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. Select **OK** to confirm the change.

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

If you want to use an external configuration file identified by the db2.jcc.propertiesFile property for your Db2® Type 2 Universal JDBC Driver, then take the following steps:

1. Browse to **Servers** > **Server Types** > **WebSphere application servers**
2. Select the appropriate server from the list
3. In the **Server Infrastructure** pane, expand **Java and Process Management**
4. Select the **Process definition** link
5. In the **processType** pane, take the following steps for each item in the list (Adjunct, Control, and Servant):
    1. Select the **processType** link
    2. In the Additional Properties pane, select the **Java Virtual Machine** link
    3. In the Additional Properties pane, select the **Custom Properties** link
    4. Select **New** and set the property as follows:

       Name: db2.jcc.propertiesFile

       Value: fully qualified name of the property file

       Click **OK** to add the property.

       See the information in Configuring IBM® WebSphere® Application Server for z/OS® on page 17 on how to set up the property file.

Save the master configuration

Click the **Save** link in the **Message(s)** box to save the master configuration. This box is displayed only after configuration changes are made.

Configuring administration security

Configure administration security by using the user registry.

About this task

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

Procedure

1. Browse to **Security** > **Global security**.
2. Set the **Available realm definitions** to be **Federated repositories** and select **Configure**.
3. Set the **Primary administrative username** to be **websphere**.
4. Select **Automatically generated server identity**.

5. Select **Ignore case for authorization** and select **OK**.
6. Enter the password for the default administrative user, for example: **websphere**, enter the confirmation and select **OK** 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. Select **Apply** to confirm the changes.
12. Select **Security** > **Global security**.
13. Expand **Web and SIP Security** and select **Single sign-on (SSO)**.
14. Select **Requires SSL**.
15. Select **OK** to confirm the change.
16. Select **Security** > **Global Security**.
17. Select the **Custom Properties** link.
18. Select **New** and set the name and value as follows:

    Name: **com.ibm.ws.security.web.logoutOnHTTPSessionExpire**

    Value: **true**
19. Select **OK** to add the new property.
20. Select **New** and set the name and value as follows:

    Name: **com.ibm.ws.security.addHttpOnlyAttributeToCookies**

    Value: **true**
21. Select **OK** to confirm the change.
22. Save the changes to the master configuration.

Restarting the application server

Restart the IBM® WebSphere® Application Server for z/OS® address spaces so that the security changes take effect and required users are added.

The IBM® WebSphere® Application Server for z/OS® address spaces must be restarted for the security changes to take effect and to add required users. The address spaces can be stopped by using the appropriate *stopServer.sh* script in the *profiles/default/bin* directory of the application server installation or by using the IBM® z/OS® operator **STOP** command appropriate for your installation.

Before you restart the application server, you must 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.

Generate the *CryptoConfig.jar* file by running the Ant **configtest** target as follows,

```
build configtest -Dcrypto.ext.dir=filedir
```

Copy the *CryptoConfig.jar* from the generated location into the Java® *jre/lib/ext* directory. If you want to customize the Cúram cryptographic configuration, see the *Cúram Security Handbook* for more information.

For sites with a read-only WebSphere® Application Server for z/OS® installation file system see the procedure Alternative crypto JAR file locations on page 16.

Start the application server by using the *startServer.sh* script in the *profiles/default/bin* directory of the WebSphere® Application Server for z/OS® installation or the IBM® 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 are made, you can now use the SDEJ scripts to restart the application server. See Starting and stopping WebSphere® Application Server for z/OS® on page 22, 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, log in to the console with the user name *websphere* and password *websphere* that you configured previously.

Testing the IBM® Db2® for z/OS® connections

When the application server restarts, test the IBM® Db2® for z/OS® connections.

Procedure

1. Browse to **Resources** > **JDBC** > **Data Sources**.
2. Check **curamdb DataSource** and/or **curamsibdb DataSource** check box.
3. Click **Test Connection**.
4. The following messages are displayed if successful:

   ```
   Test Connection for DataSource <DataSource name> on
   server <server name> at node <node name> was successful.
   ```
5. Check the WebSphere® Application Server for z/OS® logs for details of the failure, correct, and retry.

Configuring users

As detailed in Configuring security on page 18, the configured IBM® WebSphere® Application Server for z/OS® user registry is used for authentication of administrative users and the database user.

About this task

Add the WebSphere® Application Server for z/OS® administrative users and the database user to the user registry. Take the following steps:

Procedure

1. Browse to **Users and Groups** > **Manage Users**.
2. Select **Create**.
3. Complete the details for the administrative user and select **Create**.
4. Repeat the steps for the database user.

Results

If WebSphere® Application Server for z/OS® administrative security was enabled when you created the profile that the administrative user might already be defined in the registry.

Setting 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 steps for each of these configurations.

Adding the login module

Use the administration console to add the login module.

1. Browse 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. Configure the login module for the DEFAULT, WEB_INBOUND and RMI_INBOUND aliases as follows:
    1. Select **New** to configure a new Login Module.
    2. Set the **Module class name** field to be `curam.util.security.CuramLoginModule`.
    3. Select the **Use login module proxy** option.
    4. Select REQUIRED in the **Authentication strategy** field.
    5. Click **OK** to confirm the addition of the new login module.
4. Select the newly added `curam.util.security.CuramLoginModule` from the list.
5. Select the **Custom properties** link under the **Additional Properties** heading.
6. Select **New** to add the required properties that are listed in table 1.

Table 5: CuramLoginModule Custom Properties

Name	Example Value	Description
exclude_usernames	websphere, db2admin	Required. A list of user names to be excluded from authentication. This list must include the IBM® WebSphere® Application Server for z/OS® administration user (as specified in Configuring administration security on page 31) and the database user (as specified in Creating the data source login alias on page 27). The default delimiter is a comma, but might be overridden by exclude_usernames_delimiter. Any users who are listed here must be defined in the WebSphere® Application Server for z/OS® user registry.

Name	Example Value	Description
exclude_usernames_delimiter	|	*Optional*. A delimiter for the list of user names provided in exclude_usernames. A delimiter other than the default comma can be useful when user names contain embedded commas as with LDAP users.
login_trace	true	*Optional*. This property must be set to `true` to debug the authentication process. If set to `true` the invocation of the login module results in tracing information that is added to the *`SystemOut.log`* file.
module_name	DEFAULT, WEB_INBOUND or RMI_INBOUND	*Optional*. This property must 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.
check_identity_only	true	*Optional*. If this property is set to `true` the login module does not perform the usual authentication verifications. Instead, it ensures 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	*Optional*. This property is used to override the behavior of by-passing the WebSphere® Application Server for z/OS® user registry. If this property is set to true the user registry is queried during the authentication process. If this property is set to false, the user registry is not queried. **Note:** If you are specifying identity only and you are using LDAP, you might need to take more configuration steps. For more information, see Configuring identity only and LDAP on page 18.
user_registry_enabled_types	EXTERNAL	*Optional*. This property is used to specify a comma-delimited list of external user types that is processed against the user registry (for example, LDAP). For more information, see User registry on page 20 for information on the processing of the user registry.
user_registry_disabled_types	EXTGEN,EXTAUTO	*Optional*. This property specifies a comma-delimited list of external user types that is not processed against the user registry (for example, LDAP). See User registry on page 20 for more information.

7. Select **OK** to confirm the addition of the new login module.

Reordering the login module

Use the administration console to reorder the login module.

1. Browse 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 must be reordered for the DEFAULT, WEB_INBOUND and RMI_INBOUND aliases.
4. Select the **JAAS login modules** link under the **Additional Properties** heading.
5. Select **Set Order**.
6. Select **curam.util.security.CuramLoginModule** and select **Move Up**. Repeat this step until the CuramLoginModule entry is the top entry in the list.
7. Select **OK** to confirm the modifications to the order.

Disabling cross-cluster authentication

The *com.ibm.ws.security.webChallengeIfCustomSubjectNotFound* 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 without being asked for security credentials.

1. Browse to **Security** > **Global security**.
2. Select **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. Select **OK** to confirm the addition.

Save the master configuration

Click the **Save** link in the **Message(s)** box to save the master configuration. This box is displayed only after configuration changes are made.

Configuring the server

Configure server settings such as JNDI lookup and ClassLoader settings.

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 the JNDI lookup port

Procedure

1. Browse to **Servers** > **Server Types** > **WebSphere application servers**.
2. Select the relevant server from the list.
3. Expand **Ports** in the **Communications** box and select **Details**.
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 that are made to the master configuration.

Configure ClassLoader settings

Procedure

1. Browse 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 that are made to the master configuration.

Configure ORB to pass by reference

Procedure

1. Browse 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. Select **OK** to apply changes.
6. **Save** the changes that are made to the master configuration.

Configure the Java virtual machine

Procedure

1. Browse to **Servers** > **Server Types** > **WebSphere application servers**.
2. Select the appropriate server from the list.
3. In the Server Infrastructure pane, expand **Java and Process Management**.
4. Select the **Process definition** link.
5. In the **processType** pane, perform the following steps for each item in the list (Adjunct, Control, and Servant):
   a) Select the **processType** link.
   b) In the Additional Properties pane, 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** pane, select the **Custom Properties** link.
   e) Select **New** and set the properties as follows:

      - **Name**: `com.ibm.websphere.security.util.authCacheCustomKeySupport`
      - **Value**: `false`

Select **OK** to add the property.

6. **Save** the changes that are made to the master configuration.

Configuring the timer service

Procedure

1. Browse to **Servers** > **Server Types** > **WebSphere application servers**.
2. Select the appropriate server from the list.
3. In the **Container Settings** pane, expand **EJB Container Settings**.
4. Select the **EJB timer service settings** link.
5. In the **Scheduler Type** pane, 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 you created in Creating the data source login alias on page 27.
7. Select **OK** to confirm the changes.
8. **Save** the changes that are made to the master configuration.

Set up the port access

Procedure

1. Browse 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. Select **New** and set the following fields for the Client TCP/IP port:
   - **User-defined Port Name**: CuramClientEndPoint
   - **Host**: *
   - **Port**: <*client port*>

   Set <*client port*> to match the value of the property curam.client.httpport in your `AppServer.properties` file.
6. Select **OK** to apply the changes.
7. Select **New** 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.
8. Select **OK** to apply the changes.
9. Browse to **Servers** > **Server Types** > **WebSphere application Servers**.

10. Select the relevant server from the list.
11. Expand the **Web Container Settings** branch in the **Container Settings** section.
12. Select the **Web container transport chains** link.
13. Select **New** and set the following fields for the Client transport chain:
    - **Name**: `CuramClientChain`
    - **Transport Chain Template**: `WebContainer-Secure`
    - Select **Next**
    - **Use Existing Port**: `CuramClientEndPoint`
    - Select **Finish**
14. Select **New** and set the following fields for the WebServices transport chain:
    - **Name**: `CuramWebServicesChain`
    - **Transport Chain Template**: `WebContainer`
    - Select **Next**
    - **Use Existing Port**: `CuramWebServicesEndPoint`
    - Select **Next** > **Finish**.
15. Select the newly created **CuramClientChain**.
16. Select the **HTTP Inbound Channel** link.
17. Ensure that the **Use persistent keep-alive connections** check-box is checked.
18. Select **OK** to confirm the addition.
19. Browse to **Environment** > **Virtual hosts**.
20. Select **New** and add a `Virtual Host` by setting **Name** = *client_host*. Repeat this step by using the replacing *client_host* with *webservices_host*.
21. Select the **client_host** link from the list of virtual hosts.
22. Select the **Host Aliases** link in the **Additional Properties** box.
23. Select **New** to add an `Alias` by setting the following fields:
    - **Host Name** = *
    - **Port** = *<client port>*
24. 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 (for example, webservices_host)
25. Select **OK** to confirm the addition.
26. Save the changes to the master configuration as described in Disabling cross-cluster authentication on page 36.

Configuring session security integration

Procedure

1. Browse to **Servers** > **Server Types** > **WebSphere application servers**.
2. Select the relevant server from the list.
3. Select **Session management** in the **Container Settings** section
4. Deselect **Security integration**.
5. Select **OK** to apply changes.
6. **Save** the changes that are made to the master configuration.

Note:

This setting is required for Cúram web applications.

Configuring the Service Integration Bus

Configure the Service Integration Bus (SIB). A SIB supports applications with message-based and service-oriented architectures.

Procedure

1. Browse to **Service integration** > **Buses**.
2. Select **New** and set the following field **Name**: CuramBus
3. Leave all other fields as default and select **Next**.
4. Entering the **Configure bus security** wizard, Step 1.1, select **Next**
    1. In **Step 1.2** of the **Configure bus security** wizard, take the default setting and select **Next**.
    2. In **Step 1.3** of the **Configure bus security** wizard, take the default setting, and select **Next**.
    3. In **Step 1.4** of the **Configure bus security** wizard, review your settings and select **Next**.
5. In Step 2 select **Finish** to apply the changes.
6. Select the **CuramBus** now displayed on the list of Buses.
7. Select **Bus members** in the **Topology** list.
8. Select **Add** to open the **Add a New Bus Member** wizard.
9. Select the server to add to the Bus and select **Next**.
10. Select **Data store** and select **Next**.
11. 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 user name.
    - Deselect the **Create tables** option.
    - Leave all other options as the default and select **Next**.
12. Accept the default tuning parameters and select **Next**.
13. Select **Finish** to complete and exit the wizard.
14. Browse to **Service integration** > **Buses**.
15. Select the **CuramBus** now displayed on the list of Buses.
16. Select **Security** in the **AdditionaThis will open the configuration screen.l Properties** section.
17. Select **Users and groups in the bus connector role** in the **Authorization Policy** section.
18. Select **New** to open the **SIB Security Resource Wizard**.
19. Select **The built in special groups**, and select **Next**.
20. Select the **Server** and **AllAuthenticated** check boxes and select **Next**.
21. Select **Finish** to complete and exit the wizard.

22. Save the changes to the master configuration as described in Disabling cross-cluster authentication on page 36.

Configuring JMS

Configure JMS features such as connection factories, queues and topics.

Configuring the JMS connection factories

Procedure

1. Browse to **Resources** > **JMS** > **JMS providers**. The appropriate scope where the JMS resources are to be defined are selected.
2. Select the **Default messaging provider** link;
3. Select the **Connection factories** link in the **Additional Properties** box;
4. Select **New** 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 (for example, <*SERVERNAME*> /dbadmin)
   - **Mapping-configuration alias**: DefaultPrinicipalMapping
   - **Container-managed authentication alias**: Same as for the Authentication alias for XA recovery.
   - Leave all other fields as default and select **OK** to apply the changes.
5. Select **New** and set the following fields:
   1. **Name**: CuramTopicConnectionFactory
   2. **JNDI Name**: jms/CuramTopicConnectionFactory
   3. **Description**: The factory for all connections to the application queues.
   4. **Bus Name**: CuramBus
   5. **Authentication alias for XA recovery**: Same as for the `jdbc/curamdb` data source (for example, <*SERVERNAME*>/dbadmin)
   6. **Mapping-configuration alias**: DefaultPrinicipalMapping
   7. **Container-managed authentication alias**: Same as for the `jdbc/curamdb` data source (for example, *<SERVERNAME>/dbadmin)*

   Leave all other fields as default and select **OK** to apply the changes.
6. Save the changes to the master configuration as described in Disabling cross-cluster authentication on page 36.

Results

Note: Using these manual configuration steps, you cannot correctly configure security for the Cúram queue and topic connection factories. To complete this part of the configuration, you must use the `wsadmin` tool, exit the administrative console, and follow these steps:

1. Identify the queue and topic connection factory entries in the IBM® WebSphere® Application Server for z/OS® configuration *`resources.xml`* file. This file is 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, by 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, find the `<factories>` entities for the `CuramQueueConnectionFactory` and `CuramTopicConnectionFactory` and make note of the ID for each that begins `J2CConnectionFactory_` followed by a numeric (for example, 1264085551611).
2. Start the `wsadmin` WebSphere® Application Server for z/OS® script. In these examples the language is JACL, so the *-lang jacl* argument might need to be specified along with login credentials, etc. depending on your local configuration.
3. In `wsadmin` start the following commands; again, assuming node-scope definitions, a cell name of `MyNodeCell`, and a node name of `MyNode`, the resource IDs are different in your environment:
   1. Get the node and cell identifier: `$AdminConfig getid /Node:MyNode`
   2. Using the node and cell identifier from the previous step, combine it and the connection factory identifier you obtained to display the connection factory: `$AdminTask showSIBJMSConnectionFactory CuramQueueConnectionFactory(cells/MyNodeCell/nodes/MyNode|resources.xml#J2CConnectionFactory_1264085551611)`

      From the command output, 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}
      ```
   3. To set the authDataAlias use the same connection factory information, for example: `$AdminTask modifySIBJMSConnectionFactory CuramQueueConnectionFactory(cells/MyNodeCell/nodes/MyNode|resources.xml#J2CConnectionFactory_1264085551611) {-authDataAlias crouch/databaseAlias}`
   4. Save the changes: `$AdminConfig save`
   5. You can start the `showSIBJMSConnectionFactory` command to verify the change.
   6. Repeat the steps for the `CuramTopicConnectionFactory`.
   7. Exit the wsadmin session by using the `quit` command, ensuring you save your

Configuring the required queues

About this task

Use the administrative console to carry out these steps, substituting <*QueueName*> (without the angle brackets) with each of the following queue names: DPEnactment, DPError, CuramDeadMessageQueue, WorkflowActivity, WorkflowEnactment, and WorkflowError.

Procedure

1. Browse to **Service integration** > **Buses** > **CuramBus**;
2. Select the **Destinations** link in the **Destination resources** box;
3. Select **New** to open the “Create new destination” wizard.
4. Select **Queue** as the destination type and click **Next**:
5. Set the following queue attribute: **Identifier**: SIB_ <*QueueName*>

   Leave all other attributes as default and select **Next**.
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.
9. Use the **Specify** button and the entries in table 1 to set the Exception Destination and associated text filed:

   Table 6: Exception Destination Settings

Queue Name	Exception Destination
SIB_CuramDeadMessageQueue	System
SIB_DPEnactment	SIB_DPError
SIB_DPError	SIB_CuramDeadMessageQueue
SIB_WorkflowActivity	SIB_WorkflowError
SIB_WorkflowEnactment	SIB_WorkflowError
SIB_WorkflowError	SIB_CuramDeadMessageQueue

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

    Name: <*QueueName*>

    JNDI Name: jms/ <*QueueName*>

    Bus Name: CuramBus

    Queue Name: SIB_ <*QueueName*>

    Delivery Mode: Persistent

Leave all other fields as default and select **OK** to apply the changes.

Results

Save the changes to the master configuration as described in Disabling cross-cluster authentication on page 36.

Configuring the required topics

Procedure

1. Browse to **Resources** > **JMS** > **JMS providers**.
2. Select the **Default messaging provider** link.
3. Select the **Topics** link in the **Additional Properties** box.
4. Select **New** 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 all other fields default and select **OK** to apply the changes.
5. Save the changes to the master configuration as described in Disabling cross-cluster authentication on page 36.

Configuring the queue activation specifications

About this task

Take the following steps, substituting <*QueueName*>, without the angle brackets, with each of the following queue names: DPEnactment, DPError, CuramDeadMessageQueue, WorkflowActivity, WorkflowEnactment, and WorkflowError.

Procedure

1. Browse 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 specification by selecting **New** and setting 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 (for example, <*SERVERNAME*>/dbadmin)

   Leave all other fields as default and select **OK** to add the port.

Results

Save the changes to the master configuration as described in Disabling cross-cluster authentication on page 36.

Configuring the topic activation specifications

Add a new Activation Specification.

Procedure

1. 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 all other fields as default and select **OK** to apply the changes.
3. Save the changes to the master configuration as described in Disabling cross-cluster authentication on page 36.

Post configuration tasks

Create the database tables that are required for the service Integration Bus, then create the database tables that are required for the Timer Service.

Service Integration Bus database tables

IBM® 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 running 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, for example `db2`.
- *platform* is the operating system, for example `zos`.
- *username* is the user name that is required to access the database, as specified in the *`Bootstrap.properties`* property `curam.db.username`;
- *database_name* is the name of the database, 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 outputs SQL statements to define the Service Integration Bus tables, these SQL statements must be run on the target database.

Note: There are DB2 for z/OS-specific defaults for the STOGROUP and BUFFERPOOL. For more information, see WebSphere Application Server product documentation.

Timer service database tables

After setup, you must manually create the database tables that are required for the Timer Service. IBM® WebSphere® Application Server for z/OS® provides the DDL for these tables in its *$WAS_HOME/Scheduler* directory.

The DDL files that you run are the *createTablespaceDB2ZOS.ddl* and *createSchemaDB2ZOS.ddl*, in that order.

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

Manual application deployment

Use the administration console to install an enterprise application in IBM® WebSphere® Application Server for z/OS®.

About this task

Install an application, EJB component, or web module by using the administrative console.

Note: When the installation starts, use **Cancel** to exit if the installation fails. It is not enough to move to another administrative console page without first selecting **Cancel** on an application installation page.

Procedure

1. Browse to **Applications** > **New Application**.
2. Select **New Enterprise Application**.
3. Select the appropriate button and specify the full path name of the source application file or EAR file by using **Browse** in the **Path to the new application** pane
4. Select **Next**.

   The default location for the application EAR files is: *$SERVER_DIR/ear/WAS/*
5. Select **Fast Path - Prompt only when additional information is required** in the **How do you want to install the application?** pane and select **Next**.
6. Leave the defaults as they are for step 1, **Select installation options** and select **Next**.

7. In step 2, **Map modules to servers**, for every module that is listed, select a target server or a cluster from the **Clusters and Servers** list. Tick the check box beside one or more modules and then select the server or cluster and click **Apply**.
8. For the following steps, click **Next** and on the Summary pane click **Finish** to complete the installation. This step might take a few minutes and finishes with the message *Application Curam installed successfully*.
9. Save the changes to the Master Configuration. (See Disabling cross-cluster authentication on page 36 for more details.)
10. Browse to **Applications** > **Application Types** > **WebSphere enterprise applications** and select the newly installed application.
11. Select the **Class loading and update detection** option from the **Detail Properties** section.
12. Set the **Class loader order** to be **Classes loaded with local class loader first (parent last)**.
13. Set the **WAR class loader policy** to be **Single class loader for application**.
14. Select **OK**.
15. Browse to **Users and Groups -> Manage Users**. Select **Create...** and enter a user ID, password, first name, and last name.
16. Select **Create**.
17. Return to the enterprise application. Select**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 user name and password as follows:

    Note:

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

    a) Check **Select** for the mdbuser role and select **Map Users...**.
    b) Enter an appropriate user name in the **Search String** field and select **Search**.
    c) Select the ID from the **Available:** list and select **>>** to add it to the **Selected:** list and select **OK**.
    d) Select **OK**.
18. You can now update the user RunAs role by selecting the **User RunAs roles** option from the **Detail Properties** section.
19. Enter the appropriate user name and password in the **username** and **password** fields. Check **Select** for the mdbuser role and select **Apply**.
20. Select **OK**.
21. Save the changes to the master configuration.
22. When deployment is finished, you must start the application. Browse to **Applications** > **Application Types** > **WebSphere enterprise applications**,
23. Select the check box for the newly installed application, and select **Start**. This step might take a few minutes and finishes when the application status changes to indicate that it is started.
24. Test the application deployment. For example, point a web browser at the URL for the deployed application, for example:

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

Where <*Your WebSphere host*> identifies the hostname or IP address where your WebSphere® Application Server for z/OS®

system is running and <*CuramClientEndPoint*> identifies the port that is assigned (as in Set up the port access on page 38).

Network deployment

IBM® WebSphere® Application Server for z/OS® Network Deployment offers advanced deployment services, including clustering, edge services and high availability for distributed configurations.

Customizing network deployment is outside the scope this information. For more information, see the WebSphere Application Server for z/OS IBM product documentation.

Synchronizing changes

If you are operating in a network deployment environment, ensure that WebSphere® Application Server for z/OS® synchronizes its configuration after each administration console change or Ant target change.

When you save the master configuration ensure, you manually force synchronization by using the administration console:

1. Browse to **System Administration** > **Save Changes to Master Repository**.
2. Select **Synchronize changes with Nodes**.
3. Select **Save**, the synchronization might take some time
4. Check the system and application logs for synchronization completion. These messages might vary, look for messages like the following message:

   ```
   ADMS0208I: The configuration synchronization complete for cell.
   ```

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

Configuring

Before you deploy an application, you must configure the cell by using the Deployment Manager Administration Console. 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, must be started. Start the node agent by using the IBM® 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.

When 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. Browse to **Servers** > **Server Types** > **WebSphere application servers**;
2. Check the server to be started or stopped from the list and select **Start** or **Stop** as required.

The next step in the process is to configure the federated servers. All configuration is done through the Deployment Manager administrative console. For more information, see 1.7 Manual

application server configuration on page 25. When you save the master configuration, ensure that you synchronize your changes as described in Synchronizing changes on page 48.

Setting up the system JAAS login module on page 34 details the security setup that is 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. Copy `Registry.jar` from `CuramSDEJ/lib` to the `lib` directory of the Deployment Manager installation and any federated installations.

Setting up the system JAAS login module on page 34 this security setup also requires the `CryptoConfig.jar` to be copied to the `java64/lib/ext` directory within the WebSphere Application Server installation. Copy the `CryptoConfig.jar` to the same directory structure for any other WebSphere® Application Server for z/OS® installations in the environment.

Note: Before you build the application EAR for deployment, note the *BOOTSTRAP_ADDRESS* of the server to which the EARs are installed. 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 change the curam.server.port property in your `AppServer.properties` file. Changing this property also affects the port value in the `web.xml` file when building an EAR file. For more information, see *Cúram Web Client Reference*.

Related information

Deploying on the node

Follow the instructions in Manual application deployment on page 46 to manually deploy the applications on the required server. Applications can then be started or stopped using the Deployment Manager Administration Console.

Notices

Permissions for the use of these publications are granted subject to the following terms and conditions.

Applicability

These terms and conditions are in addition to any terms of use for the Merative website.

Personal use

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of Merative

Commercial use

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of Merative.

Rights

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

Merative reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by Merative, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

MERATIVE MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

Merative or its licensors may have patents or pending patent applications covering subject matter described in this document. The furnishing of this documentation does not grant you any license to these patents.

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

Any references in this information to non-Merative 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 Merative product and use of those websites is at your own risk.

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.

The licensed program described in this document and all licensed material available for it are provided by Merative under terms of the Merative Client Agreement.

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 Merative, 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. Merative, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. Merative shall not be liable for any damages arising out of your use of the sample programs.

Privacy policy

The Merative privacy policy is available at https://www.merative.com/privacy.

Trademarks

Merative ™ and the Merative ™ logo are trademarks of Merative US L.P. in the United States and other countries.

IBM®, the IBM® logo, and ibm.com® are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide.

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.

Oracle and Java are registered trademarks of Oracle and/or its affiliates.

The registered trademark Linux® is used pursuant to a sublicense from the Linux Foundation, the exclusive licensee of Linus Torvalds, owner of the mark on a worldwide basis.

Microsoft™, Windows™, 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 company, product, and service names may be trademarks or service marks of others.