

IBM Operational Decision Manager
Version 8 Release 6

*Configuring Operational Decision
Manager on JBoss*

IBM

Note

Before using this information and the product it supports, read the information in "Notices" on page 95.

This edition applies to version 8, release 6, modification 0 of Operational Decision Manager and to all subsequent releases and modifications until otherwise indicated in new editions.

© **Copyright IBM Corporation 2012, 2014.**

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

Contents

Chapter 1. Configuring Operational Decision Manager on JBoss 1

| | |
|--|----|
| Configuring Rule Execution Server on JBoss 5.1.2 EAP | 1 |
| Before you start | 1 |
| Step 1: Selecting and applying the persistence type | 2 |
| Step 2: Restricting database user permissions | 3 |
| Step 3: Establishing the database credentials | 4 |
| Step 4: Creating a data source and connection pool | 4 |
| Step 5: Configuring security | 6 |
| Step 6: Deploying the execution unit (XU) RAR file | 6 |
| Step 7: Deploying the Rule Execution Server management EAR | 8 |
| Step 8: Creating a schema for the Rule Execution Server database | 8 |
| Creating a database schema by using the Rule Execution Server console | 8 |
| Installation Settings wizard overview | 8 |
| Opening the Rule Execution Server console | 9 |
| Step 1: Welcome to the Installation Settings wizard | 10 |
| Step 2: Choose the database schema | 10 |
| Step 3: Review the database schema | 11 |
| Step 4: The Installation Settings wizard report | 11 |
| Creating a database schema by running SQL scripts | 12 |
| Step 9: Deploying the hosted transparent decision service | 13 |
| Step 10: Verifying the configuration | 13 |
| Optional configuration steps | 14 |
| Repackaging the Rule Execution Server archive using Ant | 14 |
| Configuring Rule Execution Server in different environments | 14 |
| Rule Execution Server deployment for high availability and scalability | 15 |
| Cluster configuration and topology | 15 |
| JBoss clusters | 16 |
| Configuring Rule Execution Server on JBoss 6.1 EAP | 17 |
| Before you start | 18 |
| Step 1: Selecting and applying the persistence type | 19 |
| Step 2: Restricting database user permissions | 20 |
| Step 3: Establishing the database credentials | 20 |
| Step 4: Configuring security | 21 |
| Step 5: Creating a driver and data source | 22 |
| Installing a JDBC driver as the core module | 22 |
| Creating the data source | 23 |
| Step 6: Deploying the execution unit (XU) RAR file | 24 |
| Step 7: Deploying the Rule Execution Server management EAR | 25 |

| | |
|--|----|
| Step 8: Creating a schema for the Rule Execution Server database | 25 |
| Creating a database schema by using the Rule Execution Server console | 25 |
| Installation Settings wizard overview | 26 |
| Opening the Rule Execution Server console | 27 |
| Step 1: Welcome to the Installation Settings wizard | 27 |
| Step 2: Choose the database schema | 28 |
| Step 3: Review the database schema | 28 |
| Step 4: The Installation Settings wizard report | 29 |
| Creating a database schema by running SQL scripts | 29 |
| Step 9: Deploying the hosted transparent decision service | 30 |
| Step 10: Verifying the configuration | 31 |
| Optional configuration steps | 31 |
| Repackaging the Rule Execution Server archive using Ant | 31 |
| Configuring Rule Execution Server for different environments | 32 |
| Rule Execution Server deployment for high availability and scalability | 33 |
| Cluster configuration and topology | 33 |
| Deploying Rule Execution Server in a cluster | 34 |
| Configuring the Decision Center consoles on JBoss 5.1.2 EAP | 35 |
| Before you start | 35 |
| Step 1: Restricting database user permissions | 37 |
| Step 2: Creating a data source and connection pool | 37 |
| Step 3: Configuring security on JBoss | 39 |
| Defining groups and permissions | 39 |
| Declaring custom groups | 40 |
| Step 4: Deploying the Decision Center EAR on JBoss | 41 |
| Step 5: Verifying the deployment of the Decision Center consoles | 41 |
| Step 6: Completing your Decision Center configuration | 42 |
| Completing the configuration from the Decision Center Enterprise console | 42 |
| Installation Settings wizard overview | 42 |
| Step 1: Configure the database | 43 |
| Step 2: Set up message files | 44 |
| Step 3: Set up groups | 45 |
| Step 4: Set the persistence locale | 45 |
| Step 5: Set configuration parameters | 46 |
| Completing the configuration by using Ant tasks | 47 |
| Setting up the Ant tasks environment | 47 |
| Creating the database schema | 49 |
| Defining and uploading message files | 53 |
| Setting the persistence locale | 54 |

| | |
|---|----|
| Adding or removing configuration parameters | 54 |
| Repackaging the Decision Center archive | 56 |
| Configuring the search function of the Decision Center Business console | 56 |
| Using a remote Solr search engine | 56 |
| Setting parameters for the Solr search engine | 57 |
| Additional steps to configure Decision Validation Services | 58 |
| Before you start | 58 |
| Step 1: Creating Decision Warehouse database resources | 59 |
| Step 2: Deploying the Decision Validation Services archive for the first time | 61 |
| Step 3: Checking the availability of Decision Validation Services | 61 |
| Step 4: Packaging Decision Validation Services by using Ant | 62 |
| Configuring the Decision Center consoles on JBoss 6.1 EAP. | 63 |
| Before you start | 63 |
| Step 1: Restricting database user permissions | 64 |
| Step 2: Creating a data source and connection pool | 65 |
| Step 3: Configuring security on JBoss. | 66 |
| Defining groups and permissions | 66 |
| Declaring custom groups | 67 |
| Step 4: Deploying the Decision Center EAR on JBoss | 68 |
| Step 5: Verifying the deployment of the Decision Center consoles | 68 |
| Step 6: Completing your Decision Center configuration | 69 |
| Completing the configuration from the Decision Center Enterprise console | 69 |
| Installation Settings wizard overview | 69 |
| Step 1: Configure the database | 70 |
| Step 2: Set up message files | 71 |

| | |
|---|----|
| Step 3: Set up groups | 72 |
| Step 4: Set the persistence locale | 72 |
| Step 5: Set configuration parameters | 73 |
| Completing the configuration by using Ant tasks. | 74 |
| Setting up the Ant tasks environment. | 74 |
| Creating the database schema | 76 |
| Defining and uploading message files | 80 |
| Setting the persistence locale | 81 |
| Adding or removing configuration parameters | 81 |
| Repackaging the Decision Center archive | 83 |
| Configuring the search function of the Decision Center Business console | 83 |
| Using a remote Solr search engine. | 83 |
| Setting parameters for the Solr search engine | 84 |
| Additional steps to configure Decision Validation Services | 85 |
| Before you start | 85 |
| Step 1: Creating Decision Warehouse database resources | 86 |
| Step 2: Deploying the Decision Validation Services archive for the first time | 88 |
| Step 3: Checking the availability of Decision Validation Services | 88 |
| Step 4: Packaging Decision Validation Services by using Ant | 89 |

| | |
|--|-----------|
| Chapter 2. Verifying your configuration of Decision Center. | 91 |
| Publishing a project to Decision Center | 91 |
| Opening the Decision Center consoles | 92 |
| Notices | 95 |
| Trademarks | 97 |
| Index | 99 |

Chapter 1. Configuring Operational Decision Manager on JBoss

Operational Decision Manager supports JBoss 5.1.2 EAP and 6.1 EAP.

Configuring Rule Execution Server on JBoss 5.1.2 EAP

To use Rule Execution Server on a new instance of JBoss, you must establish your database, deploy the provided archives for this server, and perform a number of configuration tasks.

Before you start

To configure Rule Execution Server on JBoss you must follow a series of steps

The first two steps concern your choice of persistence and permissions on the database. These steps are meant for a database administrator. Then, steps 3 to 5 depend on whether:

- You want to create an empty database for Rule Execution Server before or during the deployment of the WAR file.
- You want to create or clear the database schema using the Installation Settings wizard in the Rule Execution Server console, or using the SQL scripts directly.







There are also some “Optional configuration steps” on page 14.

This configuration guide presents generic instructions for all supported databases. To help beginner users, the examples are given for the embedded Derby database.

Note: Make sure to clean up prior installations. Delete the /work and /tmp directories from the folder in which you deploy Rule Execution Server: <JBOSS_HOME>/server/default. This ensures that JSP pages are recompiled during the installation.

What steps to follow

The following table summarizes the steps that you follow to configure Rule Execution Server on JBoss depending on the persistence type that you choose.

| Installation steps | Persistence | | |
|---|---|---|---|
| | File | Data source | JDBC |
| “Step 1: Selecting and applying the persistence type” on page 2 |  | Default persistence mode |  |
| “Step 2: Restricting database user permissions” on page 3 | Not applicable |  |  |
| “Step 3: Establishing the database credentials” on page 4 | Not applicable |  |  |

| Installation steps | Persistence | | |
|--|---|----------------|----------------|
| | File | Data source | JDBC |
| “Step 4: Creating a data source and connection pool” on page 4 | Not applicable | | Not applicable |
| “Step 5: Configuring security” on page 6 | | | |
| “Step 6: Deploying the execution unit (XU) RAR file” on page 6 | | | |
| “Step 7: Deploying the Rule Execution Server management EAR” on page 8 | | | |
| “Step 8: Creating a schema for the Rule Execution Server database” on page 8 | “Creating a database schema by using the Rule Execution Server console” on page 8 | Not applicable | |
| | “Creating a database schema by running SQL scripts” on page 12 | Not applicable | |
| “Step 9: Deploying the hosted transparent decision service” on page 13 | Optional | Optional | Optional |
| “Step 10: Verifying the configuration” on page 13 | Recommended | Recommended | Recommended |

Related concepts:

Troubleshooting Rule Execution Server on JBoss

Step 1: Selecting and applying the persistence type

You can change the default datasource RuleApp and Java™ XOM persistence settings by running an Ant script that generates a new Rule Execution Server management archive.

Typically, you do this if you are in development mode. This step does not apply to beginners who work with the embedded Derby database.

For you to change the persistence settings, the distribution provides an Ant script in the `<ODM_InstallDir>/executionserver/bin/ressetup.xml` file. Use it to create a new instance of the Rule Execution Server management archive and, in the case of a Java EE application server, the execution unit (XU).

Solaris users

If you use file-based persistence on Solaris, your file system must support all characters used in directory and file names that are present in the ruleset path (RuleApp name and ruleset name). Set the **LANG** system property with the encoding that is compatible with your package and rule names, for example `en_US.UTF-8`.

Decision Warehouse

If you select the file persistence type for RuleApps, you cannot use Decision Warehouse.

MySQL persistence

If you choose to use MySQL as a persistence back end, add or set the following properties in the MySQL configuration file: `my.ini` on Windows or `my.cnf` on UNIX operating systems:

```
sql-mode=STRICT_ALL_TABLES
max_allowed_packet=1073741824
```

For more information about these settings, see the MySQL 5.0 reference manual: 5.1.7. Server SQL Modes and 5.1.4 Server System Variables.

Step 2: Restricting database user permissions

If Rule Execution Server data is stored in a database, the database administrator might require that you provide the specific permissions to access the database.

Note: This step applies when database access needs to be restricted. If you manage the database yourself (for example, you use an embedded database for test purposes) or if you do not need further restrictions, skip this step and proceed to the next configuration step.

Connection to the Rule Execution Server database, as established in the data source credentials, and any subsequent requests to the database are handled through a database user. This database user (name and password), for example `resdbUser`, is defined by the database administrator and has no relation to the standard Rule Execution Server groups.

The following table gives the typical list of permissions that the database administrator must define on the Rule Execution Server database, with attention given to the type of operations. Some supported databases do not require all these permissions.

| Database permission | Operation | |
|---------------------|---------------------------------------|---|
| | Browse and edit rulesets and RuleApps | Create the Rule Execution Server schema |
| CREATE ANY INDEX | Not required | Required |
| DROP ANY INDEX | Not required | Required |
| CREATE ANY SEQUENCE | Not required | Required |
| DROP ANY SEQUENCE | Not required | Required |
| SELECT ANY SEQUENCE | Required | Not required |
| CREATE ANY TABLE | Not required | Required |
| DROP ANY TABLE | Not required | Required |
| INSERT ANY TABLE | Required | Not required |
| SELECT ANY TABLE | Required | Not required |
| UPDATE ANY TABLE | Required | Not required |
| DELETE ANY TABLE | Required | Not required |
| CREATE ANY TRIGGER | Not required | Required |

| Database permission | Operation | |
|---------------------|---------------------------------------|---|
| | Browse and edit rulesets and RuleApps | Create the Rule Execution Server schema |
| CREATE ANY VIEW | Not required | Required |
| DROP ANY VIEW | Not required | Required |

Step 3: Establishing the database credentials

You must establish the credentials of the database dedicated to Rule Execution Server if you are using database persistence.

These credentials are required to establish the datasource, which you will do in the next step.

Note: Skip to the next step if you are a beginner using the embedded Derby database.

If you set persistence to file, you can skip all the database-related tasks and proceed directly to “Step 7: Deploying the Rule Execution Server management EAR” on page 8.

If a database does not already exist, create one now by following the instructions for that database type. You can also create the database in the deployment step.

For example, if you use an XA data source, you can create an empty database with an XA data source. Add the following property to your deployment descriptor file (*-ds.xml):

```
<xa-datasource-property name="CreateDatabase">create</xa-datasource-property>
```

After the database is created, you can use the Installation Settings wizard in the Rule Execution Server console or you can run the provided SQL scripts to create the database schema so that it includes the tables and views.

Step 4: Creating a data source and connection pool

If you use data source persistence, you must create a data source and a connection pool.

Before you begin

Make sure that you have the matching JAR files in the <JBASS_HOME>/server/default/lib directory. For example, the Derby embedded database drivers require the following JAR files:

- The derby.jar file in <InstallDir>/studio/samples/shared/lib.
- The derby-plugin.jar file in <JBASS_HOME>/docs/examples/varia.
- The org.jboss.jdbc.DerbyDatabase MBean in a plug-in package in the derby-plugin.jar archive.

Procedure

To create a data source and a connection pool:

1. In the `<JBASS_HOME>/server/default/deploy` directory, create a data source descriptor file that ends with `-ds.xml`, for example `res-derby-xa-ds.xml`.

Here is an example of an XA data source for use with the Derby database. This example is based on the `derby-xa-ds.xml` file, which is provided in the `<JBASS_HOME>/docs/examples/jca` directory.

```
<?xml version="1.0" encoding="UTF-8"?>
<datasources>
  <xa-datasource>
    <jndi-name>jdbc/resdatasource</jndi-name>
    <xa-datasource-class>
      org.apache.derby.jdbc.EmbeddedXADataSource
    </xa-datasource-class>
    <xa-datasource-property name="User">
      resdbUser</xa-datasource-property>
    <xa-datasource-property name="Password">
      resdbUser</xa-datasource-property>
    <xa-datasource-property name="DatabaseName">
      c:/resdb</xa-datasource-property>
    <xa-datasource-property name="CreateDatabase">
      create</xa-datasource-property>
    <min-pool-size>5</min-pool-size>
    <max-pool-size>20</max-pool-size>
    <idle-timeout-minutes>5</idle-timeout-minutes>
    <check-valid-connection-sql>
      select count(*) from RULEAPPS
    </check-valid-connection-sql>
    <metadata>
      <type-mapping>Derby</type-mapping>
    </metadata>
  </xa-datasource>
</datasources>
```

2. Set the values of the properties in this file.

The properties to set depend on your database and data source type. In this example, these properties are enclosed between the tag pair `<xa-datasource> ... </xa-datasource>`, which identify the XA data source. The following table describes the properties.

Table 1.

| Properties | | Description |
|-------------------|-----------------------|--|
| Common properties | JNDI name | This property is independent from the type of data source and database and must be set to this value: <code><jndi-name>jdbc/resdatasource</jndi-name></code> |
| | Data source class | This property is specific to XA data sources. It defines the class that handles the connections to the database. For example: <code><xa-datasource-class>org.apache.derby.jdbc.EmbeddedXADataSource</xa-datasource-class></code> |
| | Connection properties | These properties are specific to XA data sources and are necessary for the data source to get a connection to the database. These properties depend on the database you want to connect to. For example: <code><xa-datasource-property name="DatabaseName">c:/resdb</xa-datasource-property></code> <code><xa-datasource-property name="User">resdbUser</xa-datasource-property></code> <code><xa-datasource-property name="Password">resdbUser</xa-datasource-property></code> |
| Custom properties | | Data sources can require more properties or services. For example, in the case of embedded databases (such as Derby), you must start the database first. Databases are started from a service. In the XML descriptor, this dependency is translated by the tag pair <code><depends> ... </depends></code> . |

Results

If your JBoss instance is running, it automatically detects and deploys the new descriptor.

Step 5: Configuring security

You control access to Rule Execution Server and enforce security by defining user groups and associated roles.

The main groups and their associated default user and password are summarized in the following table.

| Group | Use | Default user/password |
|-------------------|---|-------------------------|
| resAdministrators | Gives a user full administrator rights to: <ul style="list-style-type: none">• Access and use the Rule Execution Server console to populate the database schema• Deploy, browse, and modify RuleApps• Monitor the decision history, purge and back up the history• Run diagnostics and view server information | resAdmin/resAdmin |
| resDeployers | Gives a user the rights to: <ul style="list-style-type: none">• Deploy, browse, and modify RuleApps• Test rulesets | resDeployer/resDeployer |
| resMonitors | Gives a user the rights to: <ul style="list-style-type: none">• View RuleApps• Monitor decision history and access Decision Center reports | resMonitor/resMonitor |

You declare groups and users in the files `roles.properties` and `users.properties`, which you create in the directory `<JBOSS_HOME>/server/default/conf`.

Below are examples of these property files:

roles.properties

```
resAdmin=resAdministrators,resDeployers,resMonitors
resDeployer=resDeployers,resMonitors
resMonitor=resMonitors
```

users.properties

```
resAdmin=resAdmin
resDeployer=resDeployer
resMonitor=resMonitor
```

Step 6: Deploying the execution unit (XU) RAR file

Next, you deploy the resource adapter archive (RAR) for the execution unit (XU).

About this task

The execution unit (XU) is a resource adapter for Java EE application servers. It is supplied as a RAR archive that you must deploy on your application server. Deploy the XU by copying the XU RAR and XML files to the deployment directory of your application server.

In addition to the procedure described here, you also have the option to deploy the XU inside the application (embed the XU into the EAR). There might be some instances where, due to your application constraints, you have to deploy the XU inside the application. It is your decision to choose the appropriate deployment mode of the XU (either embed into the EAR or deploy as a global connector). Refer to the application server documentation for instructions on packaging a connector into an EAR.

Note:

- When the XU is deployed as a global connector, its third-party libraries (such as ASM) might be used by the deployed J2EE applications instead of the libraries deployed in the application server.
- When the XU is deployed as a global connector, use a parent `last` setting for its J2EE application if your J2EE application does not support the version of the libraries from independent software vendors that are distributed with Decision Server. If this is not possible, you might have to embed the XU into the EAR that is executing the rules.
- When using an embedded XU packaging, use a parent `last` setting for the code library if the the version of the libraries from independent software vendors that are deployed at the code library level of the application server is not compatible with the XU.

When deployed to JBoss 6.1, the XU uses the JDK logging API. The logging configuration is handled by the JBoss configuration file `jboss-logging.xml`. This file is located in your JBoss server deployment directory: `<JBOSS_HOME>/server/default/deploy`. For more information on this file, refer to the JBoss documentation.

The XU uses the logger name `com.ibm.rules.res.execution`. To write only warnings and errors to the JBoss log, add the following line in the `jboss-logging.xml` file:

```
<logger category="com.ibm.rules.res.execution">
  <level name="WARNING"/>
</logger>
```

Tip: When the default configuration in shared mode is not appropriate for your use case, you can configure Rule Execution Server so that it is scoped to a single Java™ EE application. See *Configuring Rule Execution Server in scoped mode*.

Procedure

1. In your JBoss deployment directory, `<JBOSS_HOME>/server/default/deploy`, locate the `jca-jboss-beans.xml` file.
2. Open the file in an editor and search for the text `Whether to track unclosed connections and close them`.
3. Set the **debug** attribute.

```
<property name="debug">false</property>
```
4. Save the file.

5. Copy the `jrules-res-xu-JBOSS5.rar` and `jrules-res-xu-JBOSS5-ds.xml` files from `<ODM_InstallDir>/executionserver/applicationservers/JBoss5` into the `<JBOSS_HOME>/server/default/deploy` directory.

Step 7: Deploying the Rule Execution Server management EAR

To deploy the Rule Execution Server management EAR archive, you must copy it to the deployment directory of your application server.

Procedure

1. Locate the following EAR file: `<ODM_InstallDir>/executionserver/applicationservers/JBoss5/jrules-res-management-JBOSS5.ear`
2. Copy the file to the following directory:
`<JBOSS_HOME>/server/default/deploy`

Step 8: Creating a schema for the Rule Execution Server database

After you have created a dedicated Rule Execution Server database, you can create the schema for the database by running SQL scripts from either the Rule Execution Server console or the SQL tool of your database.

Creating a database schema by using the Rule Execution Server console

To create a schema for the Rule Execution Server database, you can use the Installation Settings wizard if you work on Windows and other supported distributed platforms.

Installation Settings wizard overview:

On Windows and distributed platforms only, you can use the Installation Settings wizard of the Rule Execution Server console to choose a database type and create a schema that contains the necessary tables and views.

You can use the Installation Settings wizard to configure Rule Execution Server with database persistence.

Note: You must have created the data source connection before you use the Installation Settings wizard.

The Installation Settings wizard creates all the required tables for Rule Execution Server and Decision Warehouse.

If you are using file persistence or have an existing database schema, the Installation Settings wizard does not open when you sign in to the Rule Execution Server console. If you want to modify the database schema after having created the database tables already, you must run the SQL scripts in the database client.

The combination of persistence settings for RuleApps and managed Java XOMs affects the way in which you use the Installation Settings wizard.

- If the RuleApp persistence and the Java XOM persistence are both set to `file`, no wizard is presented.
- If the RuleApp persistence and the Java XOM persistence are both set to `datasource` or `jdbc`, the **RuleApp persistence details** part of the Installation

Settings wizard opens for you to create the schema for RuleApps and the Decision Warehouse trace when you sign in to the Rule Execution Server console. After you have completed this step, the **Java XOMs persistence details** part of the wizard opens for you to configure the database for Java XOM persistence.

- If the RuleApp persistence is set to `file` and the Java XOM is set to `datasource` or `jdbc`, you see only **Java XOMs persistence details**, and you cannot use the Decision Warehouse.
- If the RuleApp persistence is set to `datasource` and the Java XOM is set to `file` or is not defined, only **RuleApp persistence details** opens for you to create the schema for RuleApps when you sign in to the Rule Execution Server console. The wizard does not show **Java XOMs persistence details**.

The following table summarizes the cases.

| Persistence | | RuleApps | |
|-------------|------------|--|--|
| | | file | datasource/JDBC |
| Java XOMs | file | No Installation Settings wizard | RuleApps persistence details only |
| | datasource | Java XOMs persistence details only No Decision Warehouse | Complete Installation Settings wizard |

Opening the Rule Execution Server console:

To open the Rule Execution Server console and the Installation Settings wizard, you must sign in with administrator rights.

Procedure

1. Start the application server by double-clicking `run.bat` in `<JBoss_HOME>/bin`
2. Open the Rule Execution Server console in a web browser by typing `res` at the root URL on the host machine:
`http://localhost:8080/res`
 - If your browser is not running on the same host as the application server, replace `localhost` with the address of the machine.
 - If the web application is mapped to a host which is defined on a port that is different from the default of the server, change the port number to the host port number.
3. Sign in to the Rule Execution Server console as the administrator. For example:

User ID

`resAdmin`

Password

`resAdmin`

Results

If you use database persistence and the database schema is empty, the Installation Settings wizard opens and you can use it to complete the installation.

Step 1: Welcome to the Installation Settings wizard:

If you open the Rule Execution Server console with `datasource` or `jdbc` as the persistence setting and an empty database schema, the Installation Settings wizard opens.

The wizard can display the following parts:

- **RuleApp persistence details:** This part opens if you set `datasource` persistence for RuleApps, regardless of the persistence type for managed Java XOMs.
- **Java XOMs persistence details:** This part opens after **RuleApp persistence details** if you have set `datasource` or `jdbc` persistence for both RuleApps and Java XOMs. In this case, you go through the same steps twice.

The wizard starts with **Java XOMs persistence details** if you have set the persistence type to `file` for RuleApps, and to `datasource` or `jdbc` for managed Java XOMs.

Both parts of the wizard are similar, and you use them in the same way:

1.

The Welcome page provides the following information:

- Persistence details about the type of database used. This includes information about the driver and JDBC URL.
- A brief description of the purpose of the Installation Settings wizard.
- A diagnostic report that provides information about why the persistence check failed (because you have not created the database schema).

2. Click **Next** to proceed.

Step 2: Choose the database schema:

Select a schema for your database. The wizard includes settings for different databases, or you can select a customized SQL script.

About this task

You select an available Rule Execution Server database schema or you upload a custom schema.

Procedure

1. In the **Database schema selected** field, select an available type of database schema.

If you select a `db2` or `db2_os390` schema, an extra field opens so that you can enter the name of the buffer pool, which is used to create the Decision Warehouse tablespace. This buffer pool must have a page size of 32K. Check the DB2[®] documentation for information about how to create a 32K buffer pool.

Note:

The scripts for creating the Decision Warehouse database on DB2 are written for databases that use automatic storage. When you use the Installation Settings

wizard, you create both the Rule Execution Server and the Decision Warehouse database, so your database must use automatic storage.

If you have not configured your DB2 database to use automatic storage, you cannot use the Rule Execution Server console to create the Rule Execution Server tables. In this case see “Creating a database schema by running SQL scripts” on page 12.

2. Optional: Select **custom** if you want to use a customized SQL script, and then click **Browse** to select the location of the custom script.
3. Click **Next** to review the database schema.

Step 3: Review the database schema:

After you have selected a database schema, you confirm the creation of a schema for Rule Execution Server.

About this task

You can also use SQL drop statements that flush data from an existing table, and view the SQL statements. Ensure that you have a backup of your database resources.

Procedure

1. Select from the following options:

Create SQL schema “resdbUser”

Select this option to run the SQL statement for the schema type selected in the previous step.

Keep drop SQL statements

Select this option to flush data from an existing Rule Execution Server database.

Show SQL statements

Click this option to display the SQL statements.

2. Click **Execute** to start the options that you have selected.

Step 4: The Installation Settings wizard report:

After you have selected and confirmed the schema, the Installation Settings wizard reports the status of the schema creation.

Procedure

1. Click **Show execution details** to view the list of executed SQL statements.
2. Click **Finish** to open the Explorer tab in the Rule Execution Server console.

What to do next

If you have just worked in **RuleApps persistence details** and the persistence setting for managed Java XOMs is `datasource` or `jdbc`, the **Java XOMs persistence details** part of the wizard opens for you to repeat this procedure.

Creating a database schema by running SQL scripts

After you have created an empty database, you can create the Rule Execution Server database schema by running SQL scripts if you do not want to use the Installation Settings wizard.

The SQL scripts can be found in this directory: `<ODM_InstallDir>/executionserver/databases`. This directory also contains a readme file where you can find additional information about the scripts.

The script that creates the database schema is named `repository_<DatabaseName>.sql`.

Note:

If you want to use Decision Warehouse, you can also create the required database table by running the script `trace_<DatabaseName>.sql`. If you are also persisting the Java XOM in a database, you must create these tables by running the `xomrepository_<DatabaseName>.sql` script.

If you use Command Editor to run the scripts, you must log in with the credentials you use for the data source for Rule Execution Server.

Use any tool that can handle SQL to import and run the SQL scripts. The tools provided for each database include:

| Database | Database tool |
|-------------|--------------------------------|
| IBM® DB2 | DB2 command line processor |
| Derby | ij command line processor |
| MySQL | mysql command line processor |
| Oracle | sqlplus command line processor |
| Postgre SQL | Postgre SQL command line tool |
| SQL Server | Query Tool |
| Sybase | isql command line processor |

To access the database, the database user must have a user ID and a password. The database user must also have:

- complete privileges on the tables and view of the schema (create, insert, delete)
- *create index* privileges

On Oracle, the database user must also have *create trigger* and *create sequence* privileges.

When using an Oracle database, run all the scripts in the SQL Plus client.

When using DB2, the scripts that create the Rule Execution Server database tables are written for databases that use automatic storage. The following conditions apply:

- BP32K is the buffer pool that is expected in SYSCAT.BUFFERPOOLS. If BP32K is not there, you can use the existing buffer pool or create a new buffer pool named BP32K. Use the following command to query SYSCAT.BUFFERPOOLS for the existing buffer pool:

```
Select * from SYSCAT.BUFFERPOOLS
```


Otherwise, use the following command to create a buffer pool named BP32K:

```
CREATE BUFFERPOOL BP32K SIZE 2000 PAGESIZE 32K
```

- You must update the trace_db2.sql script and select the custom option in the Installation Settings wizard to run it. Modify the following line in the script to specify storage for the tablespace:

```
CREATE TABLESPACE RESDWTS PAGESIZE 32K BUFFERPOOL BP32K;
```

The following line gives an example of the tablespace specification in the script:

```
CREATE TABLESPACE RESDWTS PAGESIZE 32K MANAGED BY Database  
USING [ FILE 'C:\DB2\Container.file' 640 ] BUFFERPOOL BP32K;
```

- You might have to further modify the script based on your database settings.

Step 9: Deploying the hosted transparent decision service

You can deploy the hosted transparent decision service on your application server.

Procedure

1. Copy the following file to the deploy directory of your JBoss server:
<ODM_InstallDir>/executionserver/applicationservers/JBoss5/jrules-res-htds-JBOSS5.ear
2. Set the **ruleset.xmlDocumentDriverPool.maxSize** ruleset property to the appropriate value.
See Setting the ruleset.xmlDocumentDriverPool.maxSize property.

What to do next

For more information about checking that the hosted transparent decision service has been deployed successfully, refer to the Rule Execution Server console online help.

Step 10: Verifying the configuration

You can verify that Rule Execution Server has been successfully configured by running Rule Execution Server diagnostics.

About this task

If the diagnostics are performed before any XUs have been started, the test is passed and a message displays to verify that no XU(s) have been initialized.

Note: To let a scalable number of users access resources through the Java components, JCA assigns the task of implementing connection pooling to application server vendors.

Procedure

1. Open the Rule Execution Server console by typing res at the root URL on the host machine:
`http://localhost:8080/res`
If your browser is not running on the same host as the application server, replace localhost with the address of the machine. If the web application is mapped to a host that is defined on a different port to 8080, change the port number from 8080 to the host port number.
2. Sign in to the Rule Execution Server console.
3. Click the **Diagnostics** tab.
4. Click **Run Diagnostics**.

Optional configuration steps

You can enhance your configuration with additional options, such as multiserver configuration, or deploying onto a server cluster.

Repackaging the Rule Execution Server archive using Ant

When you need to repackage a Rule Execution Server archive to configure the Rule Execution Server, you can use an Ant task, provided that you have set up the Ant task environment.

Before you begin

Before you run the `res-setup` Ant task, you must set up the Ant task environment correctly. For more information, see [Setting up your environment to automate processes](#).

About this task

By default, persistence is set to `datasource`. To change the persistence type, you can use an Ant task. An Ant script is provided with the distribution for this purpose. The script creates new archives that use a specific persistence mode.

The following procedure repackages the archives to change the persistence mode to `file`.

Procedure

To repackage an archive file using Ant:

Write the code that creates a new XU (execution unit) RAR file and a new management EAR file that set file persistence:

```
ant -Dxu.in=./applicationserver/JBoss5/jrules-res-xu-JBOSS5.rar
-Dxu.out=myxu.rar
-Dconsole.ear.in=./applicationserver/JBoss5/jrules-res-management-JBOSS5.ear
-Dconsole.ear.out=mymanagement.ear
-Dpersistence.type=file -f ressetup.xml setup
```

Configuring Rule Execution Server in different environments

It is very likely that the development of your BRMS requires more than a single deployment of Rule Execution Server.

The development lifecycle of a business rule application is similar to any other software development process including stages for implementation, testing, deployment and maintenance. At the very least, you are likely to need an environment for your development team, one for your QA team, and another one for in-production applications. In cases where you configure Rule Execution Server in a single cell, it is good practice to isolate the rulesets that you use on each server, and ensure that the execution units (XUs) do not interfere with each other.

These instructions assume that you do not configure multiple instances of Rule Execution Server in a single cell. Use the following instructions to set up your different environments in a single cell.

To set up a data source for each environment:

1. Use a unique JNDI name. For example:
`jdbc/resdatasourceEnv1`

jdbc/resdatasourceEnv2

To configure an XU for each environment and define a J2C connection factory:

1. Depending on the version of JBoss you are running, edit the file `jrules-res-xu-JBOSS5-ds.xml`. Either use the already deployed file (in the `<JBOSS_HOME>/server/default/deploy` directory), or edit the file in the `<ODM_InstallDir>executionserver/applicationservers/JBoss5` directory.
2. Change the JNDI name to `eis/XUConnectionFactoryEnv1`. Note that execution components that invoke this XU must be modified and use this JNDI instead of the default.
3. Save the file.
4. Repeat the same process for XUs in other environments.

To deploy the Rule Execution Server console for each environment:

1. Modify the deployment descriptor of the Rule Execution Server console EAR. In the `web.xml`, uncomment the `JMX_XU_QUERY_PART` parameter and specify `xuName=xuEnv1`.
2. Deploy the Rule Execution Server console EAR on the server.
3. In the resource reference settings in the application server:
 - a. Set the JNDI name for the data source as: `jdbc/resdatasourceEnv1`.
 - b. Set the JNDI name for the XU as: `eis/ConnectionFactoryEnv1`.
4. Repeat the process to deploy the Rule Execution Server console for the other environments.

Restart the node agents after you complete the configuration.

Invoke the XU instances to register the XU with the Rule Execution Server console.

Rule Execution Server deployment for high availability and scalability

Use Rule Execution Server in a cluster or other multi-server environments.

Cluster configuration and topology:

When you deploy Rule Execution Server onto server clusters, each node hosts one execution unit (XU), which is used only by a local rule session. The cluster topology significantly affects the notification mechanism.

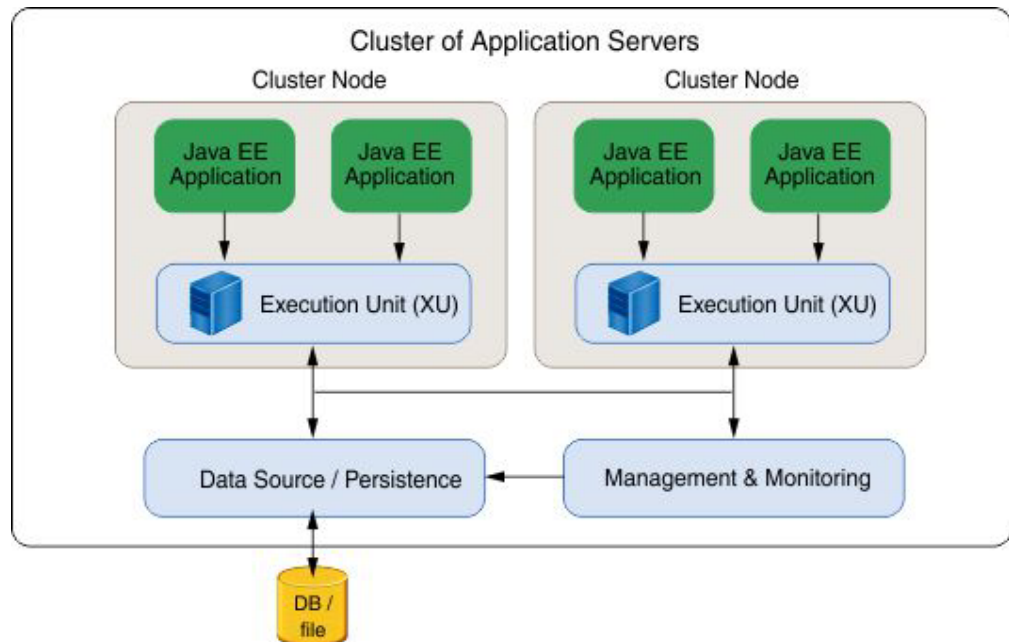
Within the Java EE framework, clusters provide mission-critical services to ensure minimal downtime and maximum scalability. A cluster is a group of application servers that transparently run your Java EE application as if it were a single entity.

Cluster implementations on Java EE application servers come with their own set of terminology. Each of the following terms are important to understand how your cluster performs:

- Cluster and component failover services
- HTTP session failover
- Single points of failure in a cluster topology

On a cluster configuration, deploy an execution unit (XU) on each node. There is one XU for each node of a cluster. Use the administration console of application servers to handle cluster deployment. A XU instance can be used only by a local (same node) rule session. The rule session and the XU communicate through direct

Java method calls, so the XU does not require serialization.



A cluster that uses Rule Execution Server involves a collaboration between the Rule Execution Server console and runtime over a TCPIP communication. The topology of the cluster has significant influence on the management of the notification mechanism when a resource is changed.

The management model is likely to use several times a basic scenario of a distributed notification mechanism within a cluster to interact with the various execution unit (XU) instances. A XU TCPIP client is deployed with the XU to collaborate with the Rule Execution Server infrastructure.

The following sequence applies:

1. A management client connects to the Rule Execution Console.
2. A notification is sent to each instance.

JBoss clusters:

When you deploy the different Rule Execution Server components on a JBoss application server, it is important to activate a single instance of the management console in a JBoss cluster.

There is no synchronization between deployed management consoles. The management console must be in the cluster to find the Execution Unit (XU) MBeans of the cluster.

Contrary to WebSphere®, the XU MBean on JBoss is activated when the RAR is deployed. Consequently, the management console automatically discovers all the XUs deployed in the cluster before any ruleset has been executed.

The notification mechanism works by transmitting messages to all the XU instances on cluster members. The management model finds the list of members in a cluster with the local MBean server, then uses the JMX remote connection on each cluster member to notify all XU instances.

To implement failover for the management console, deploy the EAR for JBoss in each `<JBOSS_HOME>/server/all/deploy-hasingleton` in the cluster. This allows you to have the management console run on a single server in the cluster. You must also specify the dependency of the high-availability singleton on the Rule Execution Server data source, by editing the file `<JBOSS_HOME>/server/all/deploy/deploy-hasingleton-service.xml`, and adding the following line:

```
<depends>jboss.jdbc:datasource=jdbc/resdatasource,service=metadata</depends>
```

to the `HASingletonController` mbean descriptor:

```
<mbean code="org.jboss.ha.singleton.HASingletonController"
  name="jboss.ha:service=HASingletonDeployer">
  ...
</mbean>
```

In this case, the management console is accessible only on the master node server in the cluster. If the master node fails, another server in the cluster is automatically selected as the master node, and the management console is activated on the server.

Use the following command to launch a server instance in a cluster:

```
$JBOSS_HOME/bin/run.bat -c all -b <server_address>
```

Where `<server_address>` is an IP address that can be reached by the other servers in the cluster.

For more information, refer to http://www.onjava.com/pub/a/onjava/2003/08/20/jboss_clustering.html

Configuring Rule Execution Server on JBoss 6.1 EAP

To use Rule Execution Server on a new instance of JBoss, you must establish your database, deploy the provided archives for this server, and perform a number of configuration tasks.

JBoss 6.1 EAP can be used in one of two operating modes: as a stand-alone server or as a managed domain.

As a stand-alone server, the operating mode is similar to the previous versions of JBoss with the option to run the server in a clustered configuration.

In a managed domain, you can run multiple instances of JBoss 6.1 EAP and manage them from a single point of control. A domain consists of a domain controller, one or more host controllers, and zero or more server groups per host.

See [About stand-alone servers and managed domains](#)

Decision Server requires a full profile. A sample full profile is provided for both stand-alone and domain modes.

The JBoss Management Console and Management CLI interfaces allow administrators to manage application deployments in a production environment. To test application deployment, developers have the option to use a file system deployment scanner, or to deploy and undeploy using Maven.

Before you start

To configure Rule Execution Server on JBoss you must follow a series of steps

The first two steps concern your choice of persistence and permissions on the database. These steps are meant for a database administrator. Then, steps 3 to 5 depend on whether:

- You want to create an empty database for Rule Execution Server before or during the deployment of the WAR file.
- You want to create or clear the database schema using the Installation Settings wizard in the Rule Execution Server console, or using the SQL scripts directly.

















There are also some “Optional configuration steps” on page 31.





This configuration guide presents generic instructions for all supported databases. To help beginner users, the examples are given for the embedded Derby database.

Note: Make sure to clean up prior installations. Delete the content of the /tmp directory in the folder in which you deploy Rule Execution Server: <JBOSS_HOME>/standalone/deployments. This ensures that JSP pages are recompiled during the installation.

What steps to follow

The following table summarizes the steps that you follow to configure Rule Execution Server on JBoss depending on the persistence type that you choose.

| Installation steps | Persistence | | |
|---|---|---|---|
| | File | Data source | JDBC |
| “Step 1: Selecting and applying the persistence type” on page 19 |  | Default persistence mode |  |
| “Step 2: Restricting database user permissions” on page 20 | Not applicable |  |  |
| “Step 3: Establishing the database credentials” on page 20 | Not applicable |  |  |
| “Step 5: Creating a driver and data source” on page 22 | Not applicable |  | Not applicable |
| “Step 4: Configuring security” on page 21 |  |  |  |
| “Step 6: Deploying the execution unit (XU) RAR file” on page 24 |  |  |  |
| “Step 7: Deploying the Rule Execution Server management EAR” on page 25 |  |  |  |

| Installation steps | | Persistence | | |
|---|--|----------------|---|---|
| | | File | Data source | JDBC |
| "Step 8: Creating a schema for the Rule Execution Server database" on page 25 | "Creating a database schema by using the Rule Execution Server console" on page 25 | Not applicable |  |  |
| | "Creating a database schema by running SQL scripts" on page 29 | Not applicable |  |  |
| "Step 9: Deploying the hosted transparent decision service" on page 30 | | Optional | Optional | Optional |
| "Step 10: Verifying the configuration" on page 31 | | Recommended | Recommended | Recommended |

Step 1: Selecting and applying the persistence type

You can change the default datasource RuleApp and Java XOM persistence settings by running an Ant script that generates a new Rule Execution Server management archive.

Typically, you do this if you are in development mode. This step does not apply to beginners who work with the embedded Derby database.

For you to change the persistence settings, the distribution provides an Ant script in the `<ODM_InstallDir>/executionserver/bin/ressetup.xml` file. Use it to create a new instance of the Rule Execution Server management archive and, in the case of a Java EE application server, the execution unit (XU).

Solaris users

If you use file-based persistence on Solaris, your file system must support all characters used in directory and file names that are present in the ruleset path (RuleApp name and ruleset name). Set the **LANG** system property with the encoding that is compatible with your package and rule names, for example `en_US.UTF-8`.

Decision Warehouse

If you select the file persistence type for RuleApps, you cannot use Decision Warehouse.

MySQL persistence

If you choose to use MySQL as a persistence back end, add or set the following properties in the MySQL configuration file: `my.ini` on Windows or `my.cnf` on UNIX operating systems:

```
sql-mode=STRICT_ALL_TABLES
max_allowed_packet=1073741824
```

For more information about these settings, see the MySQL 5.0 reference manual: 5.1.7. Server SQL Modes and 5.1.4 Server System Variables.

Step 2: Restricting database user permissions

If Rule Execution Server data is stored in a database, the database administrator might require that you provide the specific permissions to access the database.

Note: This step applies when database access needs to be restricted. If you manage the database yourself (for example, you use an embedded database for test purposes) or if you do not need further restrictions, skip this step and proceed to the next configuration step.

Connection to the Rule Execution Server database, as established in the data source credentials, and any subsequent requests to the database are handled through a database user. This database user (name and password), for example `resdbUser`, is defined by the database administrator and has no relation to the standard Rule Execution Server groups.

The following table gives the typical list of permissions that the database administrator must define on the Rule Execution Server database, with attention given to the type of operations. Some supported databases do not require all these permissions.

| Database permission | Operation | |
|---------------------|---------------------------------------|---|
| | Browse and edit rulesets and RuleApps | Create the Rule Execution Server schema |
| CREATE ANY INDEX | Not required | Required |
| DROP ANY INDEX | Not required | Required |
| CREATE ANY SEQUENCE | Not required | Required |
| DROP ANY SEQUENCE | Not required | Required |
| SELECT ANY SEQUENCE | Required | Not required |
| CREATE ANY TABLE | Not required | Required |
| DROP ANY TABLE | Not required | Required |
| INSERT ANY TABLE | Required | Not required |
| SELECT ANY TABLE | Required | Not required |
| UPDATE ANY TABLE | Required | Not required |
| DELETE ANY TABLE | Required | Not required |
| CREATE ANY TRIGGER | Not required | Required |
| CREATE ANY VIEW | Not required | Required |
| DROP ANY VIEW | Not required | Required |

Step 3: Establishing the database credentials

You must establish the credentials of the database dedicated to Rule Execution Server if you are using database persistence.

These credentials are required to establish the datasource, which you will do in the next step.

Note: Skip to the next step if you are a beginner using the embedded Derby database.

If you set persistence to file, you can skip all the database-related tasks and proceed directly to “Step 7: Deploying the Rule Execution Server management EAR” on page 25.

If a database does not already exist, create one now by following the instructions for that database type.

After the database is created, you can use the Installation Settings wizard in the Rule Execution Server console or you can run the provided SQL scripts to create the database schema so that it includes the tables and views.

Step 4: Configuring security

You control access to Rule Execution Server and enforce security by defining user groups and associated roles.

JBoss management interfaces security

The management interfaces in JBoss 6.1 EAP are secured by default to prevent security violation from remote systems. Local non-HTTP access is protected by a SASL mechanism which consists in a negotiation between the client and the server when the client connects the first time from the localhost. To access the JBoss administration console, you must register an Administrator user as follows:

1. Open a command prompt in the <JBOSS_HOME>/bin directory.
2. Enter `./add-user[.sh|.bat] <username> <password> <ManagementRealm>`
 For example: `./add-user[.sh|.bat] admin admin?12 ManagementRealm`

Rule Execution Server security

The main groups and their associated default user and password are summarized in the following table.

| Group | Use | Default user/password |
|-------------------|--|---------------------------|
| resAdministrators | Gives a user full administrator rights to: <ul style="list-style-type: none"> • Access and use the Rule Execution Server console to populate the database schema • Deploy, browse, and modify RuleApps • Monitor the decision history, purge and back up the history • Run diagnostics and view server information | resAdmin/resAdmin+1 |
| resDeployers | Gives a user the rights to: <ul style="list-style-type: none"> • Deploy, browse, and modify RuleApps • Test rulesets | resDeployer/resDeployer+1 |

| Group | Use | Default user/password |
|-------------|--|-------------------------|
| resMonitors | Gives a user the rights to: <ul style="list-style-type: none"> • View RuleApps • Monitor decision history and access Decision Center reports | resMonitor/resMonitor+1 |

You can add roles as required by Decision Server by editing the `application-roles.properties` file.

1. In Standalone mode:
`<JBOSS_HOME>/standalone/configuration/application-roles.properties`
2. In Domain mode:
`<JBOSS_HOME>/domain/configuration/application-roles.properties`

Here is an example of the `application-roles.properties` file:

```
resAdmin=resAdministrators,resDeployers,resMonitors
resDeployer=resDeployers,resMonitors
resMonitor=resMonitors
```

You can add users as required by Decision Server:

1. Open a command prompt in the `<JBOSS_HOME>/bin` directory.
2. Enter the script `./add-user[.sh|.bat] -a <username> <password>`

Here are examples based on the users listed in the table above:

- `./add-user[.sh|.bat] -a resAdmin resAdmin1+`
- `./add-user[.sh|.bat] -a resDeployer resDeployer1+`
- `./add-user[.sh|.bat] -a resMonitor resMonitor1+`

Step 5: Creating a driver and data source

If you use data source persistence, you must create a driver before creating the data source.

Installing a JDBC driver as the core module

You must install a driver before creating the data source.

Before you begin

Download the JDBC driver from your database vendor.

Procedure

1. Create a file path structure under the `<JBOSS_HOME>/modules` directory. For example, for a Derby driver, create the following directory structure:
`<JBOSS_HOME>/modules/com/ibm/derby/main.`
2. Copy the JDBC driver JAR file into the main subdirectory.
3. In the main subdirectory, create a `module.xml` file. Example for Derby:

```
<module xmlns="urn:jboss:module:1.1" name="com.ibm.derby">
  <properties>
    <property name="jboss.api" value="unsupported"/>
  </properties>
  <resources>
    <resource-root path="derby.jar"/>
    <!-- Insert resources here -->
  </resources>
```

```

    <dependencies>
      <module name="javax.api"/>
      <module name="javax.transaction.api"/>
      <module name="javax.servlet.api" optional="true"/>
    </dependencies>
  </module>

```

The module XSD is defined in the `<JBOSS_HOME>/docs/schema/module-1_2.xsd` file.

4. Start the server:

- a. Go to `<JBOSS_HOME>/bin`
- b. In stand-alone mode, run `./standalone[.sh|.bat] -c standalone-full.xml`
- c. Start the Management CLI: `<JBOSS_HOME>/bin/jboss_cli[.sh|.bat]`
- d. Enter connect.

5. To add the JDBC driver module as a driver, run the following CLI command:

```
/subsystem=datasources/jdbc-driver=DRIVER_NAME:add(driver-name=DRIVER_NAME,driver-module-name=
```

Example:

```
/subsystem=datasources/jdbc-driver=derby:add(driver-name=derby,driver-module-name=com.ibm.derby
```

Results

The JDBC driver is now installed and set up as the core module. It can be referenced by a data source.

Creating the data source

You can configure the data source by using the JBoss Management Console or the Management CLI. Here, you use the Management Console.

Procedure

1. Log in to the JBoss Management Console. By default, `http://localhost:9990/console/App.html#server-overview`.
2. Navigate to the **Datasources** panel.
 - a. Choose one of the following options.
 - In Standalone mode:
 - Click the **Profile** tab in the top right of the console.
 - In Domain mode:
 - 1) Click the **Profiles** tab in the top right of the console.
 - 2) Select the appropriate profile in the top left.
 - 3) Expand the **Subsystems** menu in the left.
 - b. Click **Connector > Datasources**
3. Select the **XA Datasources** tab.
4. Click the **Add** button at the top of the **XA Datasources** panel, and then specify the details for the new data source.
 - a. Name:
 - **Name:** RESDatasource
 - **JNDI:** java:/jdbc/resdatasource
 - b. JDBC Driver:
 - **Name:** choose the driver
 - **XA Data Source Class:** org.apache.derby.jdbc.EmbeddedXADataSource
 - c. XA Properties (at least the connection URL):

- **Name:** CreateDatabase **Value:** Create
 - **Name:** DatabaseName **Value:** /tmp/data/resdb
- d. Connection Settings:
- **Username:** the user name for a new connection. Example: RES
 - **Password:** the password for the new connection. Example: RES
 - **Security domain:** the security domain that defines the **javax.security.auth.Subject** parameter used to distinguish connections in the pool. Example: <Empty>
5. Select the newly created data source and then click the **Enable** button.

Example

You can also create the data source by using a script. Here is an example script for Derby:

```
./jboss-cli.sh --connect 'xa-data-source add --name=RESDataSource --jndi-name=java:/jdbc/resdatasource'
./jboss-cli.sh --connect '/subsystem=datasources/xa-data-source=RESDataSource/xa-datasource-properties'
./jboss-cli.sh --connect '/subsystem=datasources/xa-data-source=RESDataSource/xa-datasource-properties'
./jboss-cli.sh --connect 'xa-data-source enable --name=RESDataSource'
```

Step 6: Deploying the execution unit (XU) RAR file

Next, you deploy the resource adapter archive (RAR) for the execution unit (XU).

About this task

The execution unit (XU) is a resource adapter for Java EE application servers. It is supplied as a RAR archive that you must deploy on your application server. Deploy the XU by copying the XU RAR and XML files to the deployment directory of your application server.

In addition to the procedure described here, you also have the option to deploy the XU inside the application (embed the XU into the EAR). There might be some instances where, due to your application constraints, you have to deploy the XU inside the application. It is your decision to choose the appropriate deployment mode of the XU (either embed into the EAR or deploy as a global connector). Refer to the application server documentation for instructions on packaging a connector into an EAR.

Note:

- When the XU is deployed as a global connector, its third-party libraries (such as ASM) might be used by the deployed J2EE applications instead of the libraries deployed in the application server.
- When the XU is deployed as a global connector, use a parent last setting for its J2EE application if your J2EE application does not support the version of the libraries from independent software vendors that are distributed with Decision Server. If this is not possible, you might have to embed the XU into the EAR that is executing the rules.
- When using an embedded XU packaging, use a parent last setting for the code library if the version of the libraries from independent software vendors that are deployed at the code library level of the application server is not compatible with the XU.

Tip: When the default configuration in shared mode is not appropriate for your use case, you can configure Rule Execution Server so that it is scoped to a single Java™ EE application. See *Configuring Rule Execution Server in scoped mode*.

To deploy the XU RAR, copy the file `jrules-res-xu-JBOSS61EAP.rar` from `<ODM_InstallDir>/executionserver/applicationservers/JBoss6EAP` into the `<JBOSS_HOME>/standalone/deployments` directory.

When deployed to JBoss 6.1 EAP, the XU uses the JDK logging API.

Procedure

To configure the XU log level with the JBoss Management Console, proceed as follows:

1. Log in to the JBoss Management Console.
2. Navigate to the logging subsystem configuration.
 - Standalone server
 - a. Click **Profile** in the top right of the console.
 - b. Click **Core > Logging** on the left.
 - Managed domain
 - a. Click **Profiles** in the top right of the console.
 - b. Select the profile on the left, and then click **Core > Logging**.
3. Click the **Log Categories** tab.
4. Click the **Add** button and define the following details.
 - **Name:** `com.ibm.rules.res.execution`
 - **Log Level:** specifies which message levels are logged by this logger. Message levels lower than this value are discarded.
 - **User Parent Handler:** specifies whether or not this logger sends its output to the parent logger.
5. Save.
6. Restart the server.

Step 7: Deploying the Rule Execution Server management EAR

To deploy the Rule Execution Server management EAR archive, you must copy it to the deployment directory of your application server.

Procedure

Copy the file `jrules-res-management-JBOSS61EAP.ear` from `<ODM_InstallDir>/executionserver/applicationservers/JBoss6EAP/` to the `<JBOSS_HOME>/standalone/deployments` directory.

Step 8: Creating a schema for the Rule Execution Server database

After you have created a dedicated Rule Execution Server database, you can create the schema for the database by running SQL scripts from either the Rule Execution Server console or the SQL tool of your database.

Creating a database schema by using the Rule Execution Server console

To create a schema for the Rule Execution Server database, you can use the Installation Settings wizard if you work on Windows and other supported distributed platforms.

Installation Settings wizard overview:

On Windows and distributed platforms only, you can use the Installation Settings wizard of the Rule Execution Server console to choose a database type and create a schema that contains the necessary tables and views.

You can use the Installation Settings wizard to configure Rule Execution Server with database persistence.

Note: You must have created the data source connection before you use the Installation Settings wizard.

The Installation Settings wizard creates all the required tables for Rule Execution Server and Decision Warehouse.

If you are using file persistence or have an existing database schema, the Installation Settings wizard does not open when you sign in to the Rule Execution Server console. If you want to modify the database schema after having created the database tables already, you must run the SQL scripts in the database client.

The combination of persistence settings for RuleApps and managed Java XOMs affects the way in which you use the Installation Settings wizard.

- If the RuleApp persistence and the Java XOM persistence are both set to file, no wizard is presented.
- If the RuleApp persistence and the Java XOM persistence are both set to datasource or jdbc, the **RuleApp persistence details** part of the Installation Settings wizard opens for you to create the schema for RuleApps and the Decision Warehouse trace when you sign in to the Rule Execution Server console. After you have completed this step, the **Java XOMs persistence details** part of the wizard opens for you to configure the database for Java XOM persistence.
- If the RuleApp persistence is set to file and the Java XOM is set to datasource or jdbc, you see only **Java XOMs persistence details**, and you cannot use the Decision Warehouse.
- If the RuleApp persistence is set to datasource and the Java XOM is set to file or is not defined, only **RuleApp persistence details** opens for you to create the schema for RuleApps when you sign in to the Rule Execution Server console. The wizard does not show **Java XOMs persistence details**.

The following table summarizes the cases.

| Persistence | | RuleApps | |
|-------------|------------|--|--|
| | | file | datasource/JDBC |
| Java XOMs | file | No Installation Settings wizard | RuleApps persistence details only |
| | datasource | Java XOMs persistence details only No Decision Warehouse | Complete Installation Settings wizard |

Opening the Rule Execution Server console:

To open the Rule Execution Server console and the Installation Settings wizard, you must sign in with administrator rights.

Procedure

1. Start the application server by double-clicking `run.bat` in `<JBoss_HOME>/bin`
2. Open the Rule Execution Server console in a web browser by typing `res` at the root URL on the host machine:
`http://localhost:8080/res`
 - If your browser is not running on the same host as the application server, replace `localhost` with the address of the machine.
 - If the web application is mapped to a host which is defined on a port that is different from the default of the server, change the port number to the host port number.
3. Sign in to the Rule Execution Server console as the administrator. For example:

User ID

`resAdmin`

Password

`resAdmin1+`

Results

If you use database persistence and the database schema is empty, the Installation Settings wizard opens and you can use it to complete the installation.

Step 1: Welcome to the Installation Settings wizard:

If you open the Rule Execution Server console with `datasource` or `jdbc` as the persistence setting and an empty database schema, the Installation Settings wizard opens.

The wizard can display the following parts:

- **RuleApp persistence details:** This part opens if you set `datasource` persistence for RuleApps, regardless of the persistence type for managed Java XOMs.
- **Java XOMs persistence details:** This part opens after **RuleApp persistence details** if you have set `datasource` or `jdbc` persistence for both RuleApps and Java XOMs. In this case, you go through the same steps twice.

The wizard starts with **Java XOMs persistence details** if you have set the persistence type to `file` for RuleApps, and to `datasource` or `jdbc` for managed Java XOMs.

Both parts of the wizard are similar, and you use them in the same way:

1.

The Welcome page provides the following information:

- Persistence details about the type of database used. This includes information about the driver and JDBC URL.
- A brief description of the purpose of the Installation Settings wizard.
- A diagnostic report that provides information about why the persistence check failed (because you have not created the database schema).

2. Click **Next** to proceed.

Step 2: Choose the database schema:

Select a schema for your database. The wizard includes settings for different databases, or you can select a customized SQL script.

About this task

You select an available Rule Execution Server database schema or you upload a custom schema.

Procedure

1. In the **Database schema selected** field, select an available type of database schema.

If you select a db2 or db2_os390 schema, an extra field opens so that you can enter the name of the buffer pool, which is used to create the Decision Warehouse tablespace. This buffer pool must have a page size of 32K. Check the DB2 documentation for information about how to create a 32K buffer pool.

Note:

The scripts for creating the Decision Warehouse database on DB2 are written for databases that use automatic storage. When you use the Installation Settings wizard, you create both the Rule Execution Server and the Decision Warehouse database, so your database must use automatic storage.

If you have not configured your DB2 database to use automatic storage, you cannot use the Rule Execution Server console to create the Rule Execution Server tables. In this case see “Creating a database schema by running SQL scripts” on page 29.

2. Optional: Select **custom** if you want to use a customized SQL script, and then click **Browse** to select the location of the custom script.
3. Click **Next** to review the database schema.

Step 3: Review the database schema:

After you have selected a database schema, you confirm the creation of a schema for Rule Execution Server.

About this task

You can also use SQL drop statements that flush data from an existing table, and view the SQL statements. Ensure that you have a backup of your database resources.

Procedure

1. Select from the following options:

Create SQL schema “resdbUser”

Select this option to run the SQL statement for the schema type selected in the previous step.

Keep drop SQL statements

Select this option to flush data from an existing Rule Execution Server database.

Show SQL statements

Click this option to display the SQL statements.

2. Click **Execute** to start the options that you have selected.

Step 4: The Installation Settings wizard report:

After you have selected and confirmed the schema, the Installation Settings wizard reports the status of the schema creation.

Procedure

1. Click **Show execution details** to view the list of executed SQL statements.
2. Click **Finish** to open the Explorer tab in the Rule Execution Server console.

What to do next

If you have just worked in **RuleApps persistence details** and the persistence setting for managed Java XOMs is `datasource` or `jdbc`, the **Java XOMs persistence details** part of the wizard opens for you to repeat this procedure.

Creating a database schema by running SQL scripts

After you have created an empty database, you can create the Rule Execution Server database schema by running SQL scripts if you do not want to use the Installation Settings wizard.

The SQL scripts can be found in this directory: `<ODM_InstallDir>/executionserver/databases`. This directory also contains a `readme` file where you can find additional information about the scripts.

The script that creates the database schema is named `repository_<DatabaseName>.sql`.

Note:

If you want to use Decision Warehouse, you can also create the required database table by running the script `trace_<DatabaseName>.sql`. If you are also persisting the Java XOM in a database, you must create these tables by running the `xomrepository_<DatabaseName>.sql` script.

If you use Command Editor to run the scripts, you must log in with the credentials you use for the data source for Rule Execution Server.

Use any tool that can handle SQL to import and run the SQL scripts. The tools provided for each database include:

| Database | Database tool |
|-------------|--------------------------------|
| IBM DB2 | DB2 command line processor |
| Derby | ij command line processor |
| MySQL | mysql command line processor |
| Oracle | sqlplus command line processor |
| Postgre SQL | Postgre SQL command line tool |

| Database | Database tool |
|------------|-----------------------------|
| SQL Server | Query Tool |
| Sybase | isql command line processor |

To access the database, the database user must have a user ID and a password. The database user must also have:

- complete privileges on the tables and view of the schema (create, insert, delete)
- *create index* privileges

On Oracle, the database user must also have *create trigger* and *create sequence* privileges.

When using an Oracle database, run all the scripts in the SQL Plus client.

When using DB2, the scripts that create the Rule Execution Server database tables are written for databases that use automatic storage. The following conditions apply:

- BP32K is the buffer pool that is expected in SYSCAT.BUFFERPOOLS. If BP32K is not there, you can use the existing buffer pool or create a new buffer pool named BP32K. Use the following command to query SYSCAT.BUFFERPOOLS for the existing buffer pool:

```
Select * from SYSCAT.BUFFERPOOLS
```

Otherwise, use the following command to create a buffer pool named BP32K:

```
CREATE BUFFERPOOL BP32K SIZE 2000 PAGESIZE 32K
```

- You must update the trace_db2.sql script and select the custom option in the Installation Settings wizard to run it. Modify the following line in the script to specify storage for the tablespace:

```
CREATE TABLESPACE RESDWTS PAGESIZE 32K BUFFERPOOL BP32K;
```

The following line gives an example of the tablespace specification in the script:

```
CREATE TABLESPACE RESDWTS PAGESIZE 32K MANAGED BY Database  
USING [ FILE 'C:\DB2\Container.file' 640 ] BUFFERPOOL BP32K;
```

- You might have to further modify the script based on your database settings.

Step 9: Deploying the hosted transparent decision service

You can deploy the hosted transparent decision service on your application server.

Procedure

1. Copy the file `jrules-res-htds-JBOSS61EAP.ear` from `<ODM_InstallDir>/executionserver/applicationserver/JBoss6EAP` to the `<JBOSS_HOME>/standalone/deployments` directory.
2. Set the `ruleset.xmlDocumentDriverPool.maxSize` ruleset property to the appropriate value.
See `Setting the ruleset.xmlDocumentDriverPool.maxSize` property.

What to do next

For more information about checking that the hosted transparent decision service has been deployed successfully, refer to the Rule Execution Server console online help.

Step 10: Verifying the configuration

You can verify that Rule Execution Server has been successfully configured by running Rule Execution Server diagnostics.

About this task

If the diagnostics are performed before any XUs have been started, the test is passed and a message displays to verify that no XU(s) have been initialized.

Note: To let a scalable number of users access resources through the Java components, JCA assigns the task of implementing connection pooling to application server vendors.

Procedure

1. Open the Rule Execution Server console by typing `res` at the root URL on the host machine:
`http://localhost:8080/res`
If your browser is not running on the same host as the application server, replace `localhost` with the address of the machine. If the web application is mapped to a host that is defined on a different port to 8080, change the port number from 8080 to the host port number.
2. Sign in to the Rule Execution Server console.
3. Click the **Diagnostics** tab.
4. Click **Run Diagnostics**.

Optional configuration steps

You can enhance your configuration with additional options, such as multiserver configuration, or deploying onto a server cluster.

Repackaging the Rule Execution Server archive using Ant

When you need to repackage a Rule Execution Server archive to configure the Rule Execution Server, you can use an Ant task, provided that you have set up the Ant task environment.

Before you begin

Before you run the `res-setup` Ant task, you must set up the Ant task environment correctly. For more information, see [Setting up your environment to automate processes](#).

About this task

By default, persistence is set to `datasource`. To change the persistence type, you can use an Ant task. An Ant script is provided with the distribution for this purpose. The script creates new archives that use a specific persistence mode.

The following procedure repackages the archives to change the persistence mode to `file`.

Procedure

To repackage an archive file using Ant:

Write the code that creates a new XU (execution unit) RAR file and a new management EAR file that set file persistence:

```
ant -Dxu.in=../applicationserver/JBoss6EAP/jrules-res-xu-JBOSS61EAP.rar
-Dxu.out=myxu.rar
-Dconsole.ear.in=../applicationserver/JBoss6EAP/jrules-res-management-JBOSS61EAP.ear
-Dconsole.ear.out=mymanagement.ear
-Dpersistence.type=file -f ressetup.xml setup
```

Configuring Rule Execution Server for different environments

It is very likely that the development of your BRMS requires more than a single deployment of Rule Execution Server.

The development lifecycle of a business rule application is similar to any other software development process including stages for implementation, testing, deployment and maintenance. At the very least, you are likely to need an environment for your development team, one for your QA team, and another one for in-production applications. In cases where you configure Rule Execution Server in a single cell, it is good practice to isolate the rulesets that you use on each server, and ensure that the execution units (XUs) do not interfere with each other.

These instructions assume that you do not configure multiple instances of Rule Execution Server in a single cell. Use the following instructions to set up your different environments in a single cell.

To set up a data source for each environment:

Use a unique JNDI name. For example:

- jdbc/resdatasourceEnv1
- jdbc/resdatasourceEnv2
-

To configure a XU for each environment and define a J2C connection factory:

1. Open the `<ODM_InstallDir>executionserver/applicationserver/JBoss6EAP/jrules-res-xu-JBOSS61EAP.ear` archive with a decompression tool.
2. In the `jrules-res-xu-JBOSS61EAP.ear /Meta-Inf/ironjacamar.xml` file, make the following highlighted change: `<connection-definition class-name="ilog.rules.res.xu.spi.IlrManagedXUConnectionFactory" enabled="true" jndi-name="java:/eis/XUConnectionFactoryEnv1" pool-name="eis/XUConnectionFactory" use-ccm="true" use-java-context="true">`. Execution components that invoke this XU must be modified and use this JNDI name instead of the default.
3. Save the file.
4. Repeat the same process for XUs in other environments.

To deploy the Rule Execution Server console for each environment:

1. Modify the deployment descriptor of the Rule Execution Server console EAR. In the `web.xml`, uncomment the `JMX_XU_QUERY_PART` parameter and specify `xuName=xuEnv1`.
2. Deploy the Rule Execution Server console EAR on the server.
3. In the resource reference settings in the application server:
 - a. Set the JNDI name for the data source as: `jdbc/resdatasourceEnv1`.
 - b. Set the JNDI name for the XU as: `eis/XUConnectionFactoryEnv1`.

- Repeat the process to deploy the Rule Execution Server console for the other environments.

Restart the node agents after you complete the configuration.

Invoke the XU instances to register the XU with the Rule Execution Server console.

Rule Execution Server deployment for high availability and scalability

Use Rule Execution Server in a cluster or other multi-server environments.

Cluster configuration and topology:

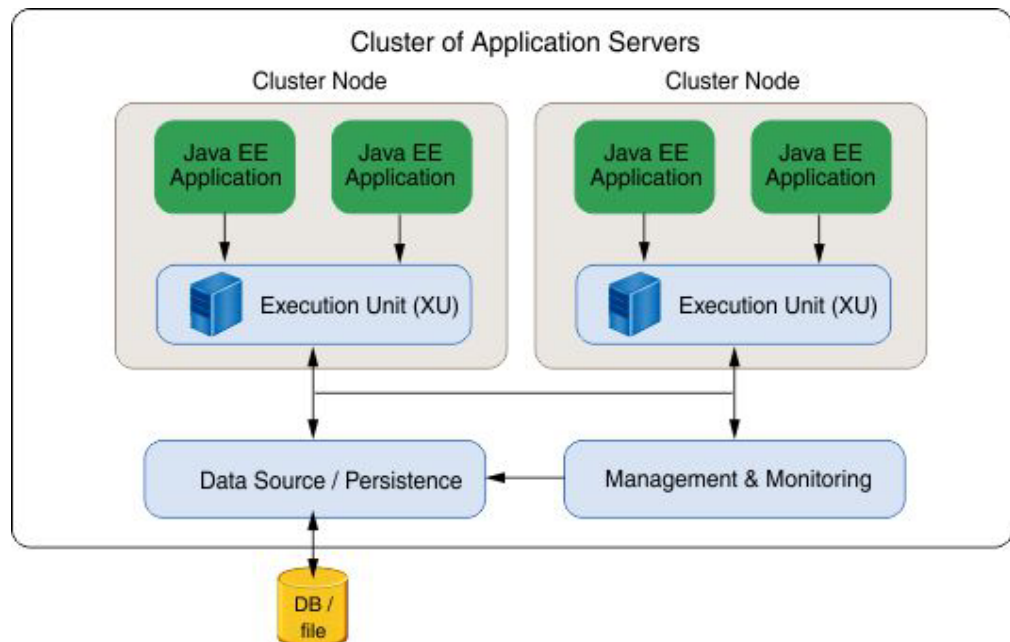
When you deploy Rule Execution Server onto server clusters, each node hosts one execution unit (XU), which is used only by a local rule session. The cluster topology significantly affects the notification mechanism.

Within the Java EE framework, clusters provide mission-critical services to ensure minimal downtime and maximum scalability. A cluster is a group of application servers that transparently run your Java EE application as if it were a single entity.

Cluster implementations on Java EE application servers come with their own set of terminology. Each of the following terms are important to understand how your cluster performs:

- Cluster and component failover services
- HTTP session failover
- Single points of failure in a cluster topology

On a cluster configuration, deploy an execution unit (XU) on each node. There is one XU for each node of a cluster. Use the administration console of application servers to handle cluster deployment. A XU instance can be used only by a local (same node) rule session. The rule session and the XU communicate through direct Java method calls, so the XU does not require serialization.



A cluster that uses Rule Execution Server involves a collaboration between the Rule Execution Server console and runtime over a TCPIP communication. The topology of the cluster has significant influence on the management of the notification mechanism when a resource is changed.

The management model is likely to use several times a basic scenario of a distributed notification mechanism within a cluster to interact with the various execution unit (XU) instances. A XU TCPIP client is deployed with the XU to collaborate with the Rule Execution Server infrastructure.

The following sequence applies:

1. A management client connects to the Rule Execution Console.
2. A notification is sent to each instance.

Deploying Rule Execution Server in a cluster:

When you deploy the different Rule Execution Server components (console and runtime) on a JBoss application server, it is important to activate a single instance of the console in a JBoss cluster.

About this task

There is no synchronization between deployed Rule Execution Server consoles.

The Rule Execution Server console automatically discovers all the XUs deployed in the cluster before any ruleset has been executed.

The notification mechanism works by transmitting messages to all the XU instances on cluster members. The management model finds the list of members in a cluster with the TCP/IP Notification protocol, then uses the TCP/IP remote connection on each cluster member to notify all XU instances.

Procedure

To deploy the Rule Execution Server in a cluster.

1. Configure the Rule Execution Server console for TCP/IP management. See https://infocenter.ilogfr-bso.fr.ibm.com/infocenter/dmanager/trunk/topic/com.ibm.odm.dserver.rules.res.managing/topics/tsk_res_rescons_config_tcpip.html
2. Configure the execution units (XU) to connect to a TCP/IP management server. See https://infocenter.ilogfr-bso.fr.ibm.com/infocenter/dmanager/trunk/topic/com.ibm.odm.dserver.rules.res.managing/topics/tsk_res_config_xu_tcpip.html
The **Hostname** parameter must refer to the host name of the server where the console is deployed.
3. Deploy the Rule Execution Server console modified in step 1 and the XU in a stand-alone instance. See steps 4, 6, 7
4. Deploy the XU with the TCP/IP notification activated in a clustered JBoss domain. Choose the targeted domain and deploy the XU with the TCP/IP activated.
5. Verify the configuration.
 - a. To open the Rule Execution Server console, type `res` at the root URL on the host machine: `http://localhost:8080/res`. If your browser is not running on the same host as the application server, replace `localhost` with the

address of the machine. If the web application is mapped to a host that is defined on a different port to 8080, change the port number from 8080 to the host port number.

- b. Sign in to the Rule Execution Server console.
- c. Click the **Diagnostics** tab.
- d. Click **Run Diagnostics**.
- e. Go to the **Server Info** tab. You should see the XUs attached to the Rule Execution Server console.

Example

You can deploy the XU and Rule Execution Server console configured with TCP/IP notification activated in the Group Member named Others. Proceed as follows:

1. Go to <JBOSS_HOME>/bin.
2. Enter ./domain.[sh|bat]
3. Open the JBoss Management Console.
4. Start the **Others Domain Group**.
5. Click the **Profiles** tab.
6. Configure the Rule Execution Server data source for the full and full-ha profiles.
7. Click the **Runtimes** tab.
8. Click **Management Deployments**.
9. Deploy the XU with TCP/IP activated.
 - a. Click **Add**.
 - b. Browse to the XU .rar file that has TCP/IP activated.
 - c. Click **Next** and then **Save**.
10. Deploy the Rule Execution Server console with TCP/IP activated. .
 - a. Click **Add** and then **Add** again.
 - b. Browse to the Rule Execution Server console EAR file with TCP/IP activated.
11. Assign the jrules-res-xu-JBOSS61EAP.rar XU to the **Other Server Group**.
12. Assign the jrules-res-xu-JBOSS61EAP.rar XU to the **Main Server Group**.
13. Assign the jrules-res-management-JBOSS61EAP.ear XU to the **Other Server Group**.
14. Verify the configuration.

Configuring the Decision Center consoles on JBoss 5.1.2 EAP

To use the Decision Center Enterprise console on a new instance of the JBoss application server, you must deploy the provided archive for this server and perform a number of configuration tasks.

Before you start

Review the steps that you must perform to complete the configuration.

These instructions are intended for users familiar with JBoss and the database they are using. Examples using the Derby embedded database are included throughout to help beginners.

Refer to “What steps to follow” to understand the steps that you must follow to configure Decision Center on your application server.

You can complete the configuration of your application server with the Installation Settings wizard in the Decision Center console or with Ant scripts.







If you need to configure Decision Validation Services, see “Additional steps to configure Decision Validation Services” on page 58.

After finishing the configuration, you can use Decision Center (see Opening the Decision Center consoles). There is no rule project the first time you open the console. You must publish a project from Rule Designer (see Publishing a project to Decision Center).

Note: If you have rule projects created with a previous product version, refer to the Migrating topics on how to upgrade the Decision Center database schema.

What steps to follow

The steps that you follow to configure Decision Center on JBoss 5 and 6.1 are summarized in the following table:

| Step | Required |
|---|--|
| “Step 1: Restricting database user permissions” on page 37 |  |
| “Step 2: Creating a data source and connection pool” on page 37 |  |
| “Step 3: Configuring security on JBoss” on page 39 | “Defining groups and permissions” on page 39  |
| | “Declaring custom groups” on page 40 Optional |
| “Step 4: Deploying the Decision Center EAR on JBoss” on page 41 |  |
| “Step 5: Verifying the deployment of the Decision Center consoles” on page 41 | Recommended |
| “Step 6: Completing your Decision Center configuration” on page 42 | “Completing the configuration from the Decision Center Enterprise console” on page 42  |
| | “Completing the configuration by using Ant tasks” on page 47 Use one of the two methods to complete the installation. Required for distributed platforms only. |
| Verifying your configuration of Decision Center |  |

Step 1: Restricting database user permissions

Decision Center data is stored in a database. The database administrator might require that you provide the specific permissions that you need when accessing this database.

Note: This step applies when database access needs to be restricted. If you manage the database yourself (for example, you use an embedded database for test purposes) or if you do not need further restrictions, skip this step and proceed to the next configuration step.

Connection to the Decision Center database, established in the data source credentials, and any subsequent requests to the database are handled through a database user. This database user (name and password), for example `rtsdbUser`, is defined by the database administrator and has no relation to the standard Decision Center groups.

The following table gives the permissions that the database administrator must define on the Decision Center database, with attention given to the type of operations that you want to perform. Some supported databases do not require all the above permissions.

| Database permission | Operation | | | |
|---------------------|---------------------|--|--|-----------------------------|
| | Use Decision Center | Create the database schema by using the Decision Center console or Ant tasks | Modify the database schema by using the Decision Center console or Ant tasks | Migrate the database schema |
| ALTER TABLE | Not required | Not required | Required | Required |
| CREATE INDEX | Not required | Required | Required | Required |
| CREATE ROLE | Not required | Not required | Not required | Required |
| CREATE SEQUENCE | Not required | Required | Required | Required |
| CREATE TABLE | Not required | Required | Required | Required |
| CREATE VIEW | Not required | Required | Required | Required |
| DROP INDEX | Not required | Not required | Required | Required |
| DROP SEQUENCE | Not required | Not required | Required | Required |
| DROP TABLE | Not required | Not required | Required | Required |
| DROP VIEW | Not required | Not required | Required | Required |
| INSERT TABLE | Required | Required | Required | Required |
| SELECT SEQUENCE | Required | Required | Required | Required |
| SELECT TABLE | Required | Required | Required | Required |
| UPDATE TABLE | Required | Required | Required | Required |

Step 2: Creating a data source and connection pool

You declare a data source to store the data that is used by Decision Center.

Before you begin

You must have a database to create a data source.

Note: Beginners who use the Derby embedded database can create the database as part of this procedure.

For a list of supported databases, see the IBM Support site.

Procedure

1. In the `<JBOSS_HOME>/server/default/deploy` directory, create an XML descriptor file that ends with `-ds.xml`, for example `dc-derby-ds.xml`.

The following `dc-derby-ds.xml` example is based on the file `derby-ds.xml` provided in `<JBOSS_HOME>/docs/examples/jca`.

```
<?xml version="1.0" encoding="UTF-8"?>
<datasources>
  <local-tx-datasource>
    <jndi-name>jdbc/ilogDataSource</jndi-name>
    <connection-url>jdbc:derby:c:/rtsdb;create=true</connection-url>
    <driver-class>org.apache.derby.jdbc.EmbeddedDriver</driver-class>
    <user-name>rtsdbUser</user-name>
    <password>rtsdbUser</password>
    <min-pool-size>5</min-pool-size>
    <max-pool-size>20</max-pool-size>
    <idle-timeout-minutes>5</idle-timeout-minutes>
    <track-statements/>
  </local-tx-datasource>
</datasources>
```

2. Set the values of the properties in this file.

By default, JBoss uses local transaction data sources that are identified by the tags: `<local-tx-datasource> ... </local-tx-datasource>`

The properties to set in the XML descriptor depend on the type of data source and on the database you want to connect to. You must set at least the following properties:

JNDI name

This property is independent from the type of data source and database and **must** be set to this value: `<jndi-name>jdbc/ilogDataSource</jndi-name>`

Connection URL

This property is for local transaction data sources. It defines the JDBC connection URL. For example: `<connection-url>jdbc:derby:c:/rtsdb;create=true</connection-url>`

Driver class

This property is for local transaction data sources. It defines the class that handles the connections to the database. For example: `<driver-class>org.apache.derby.jdbc.EmbeddedDriver</driver-class>`

Make sure that you have the equivalent `.jar` files in `<JBOSS_HOME>/server/default/lib`. For example, for the Derby embedded database drivers:

- `derby.jar`, provided in `<InstallDir>/studio/samples/shared/lib`.
- `derby-plugin.jar`, provided in `<JBOSS_HOME>/docs/examples/varia`.

User name and Password

These properties are for local transaction data sources. The tags `username="rtsdbUser"` and `password="rtsdbUser"` are used to access the database. These user name and password are used only to access the database.

Note: If the transaction isolation level is not set to `READ-COMMITTED`, Decision Center might encounter various database deadlocks. Therefore, you might have to configure data source isolation for better concurrency.

What to do next

- If you must be 100% compliant with Java EE specifications, you must disable direct JNDI data source lookup. Instructions are provided in `<ODM_InstallDir>/teamserver/lib/jrules-teamserver.jar`, in the `/ilog/rules/teamserver/preferences.properties` file.
- For Derby:
 - The database name is the directory in which the database is created.
 - The `org.jboss.jdbc.DerbyDatabase` MBean is provided in a plug-in that is packaged in the `derby-plugin.jar` archive.

Step 3: Configuring security on JBoss

You control access to Decision Center and enforce security by defining user groups and associated roles.

Defining groups and permissions

You control access to Decision Center and enforce security by defining user groups and associated roles.

About this task

Any user of Decision Center must belong to at least one of these mandatory groups:

- `rtsAdministrator`
- `rtsConfigManager`
- `rtsInstaller`
- `rtsUser`

Adherence to these groups determines what parts of Decision Center a user can access. You must create all these groups. For testing purposes, also create a default user and password for each of these groups. In addition, if you want to follow the Decision Center permissions tutorial in your own installation, you must create the two `Validator` and `Eligibility` custom groups.

The following table summarizes the mandatory and custom groups and their associated role, default user, and password.

| Group | Use | Default User - Password |
|-------------------------------|---|------------------------------------|
| <code>rtsAdministrator</code> | Mandatory, gives the user administrator access. | <code>rtsAdmin - rtsAdmin</code> |
| <code>rtsConfigManager</code> | Mandatory, gives the user configuration manager access. | <code>rtsConfig - rtsConfig</code> |
| <code>rtsUser</code> | Mandatory, gives a user standard access. | <code>rtsUser1 - rtsUser1</code> |

| Group | Use | Default User - Password |
|--------------|--|-------------------------|
| rtsInstaller | Mandatory, gives the user access to the Installation Settings wizard. | rtsAdmin - rtsAdmin |
| Validator | Optional custom group, used in the Decision Center permissions tutorial. | Val - Val |
| Eligibility | Optional custom group, used in the Decision Center permissions tutorial. | Eli - Eli |

Note: For more information on the Decision Center groups, see Groups.

You declare groups and users in the files `roles.properties` and `users.properties`.

The following procedure is a suggested configuration with a default user for each of the basic groups.

Procedure

1. Create the files `roles.properties` and `users.properties` in the configuration directory: `<JBoss_HOME>/server/default/conf`.
2. In `roles.properties`, map users to their groups.

For example:

```

rtsUser1=rtsUser
rtsConfig=rtsConfigManager, rtsUser
rtsAdmin=rtsAdministrator,rtsInstaller,rtsUser
Eli=Eligibility,rtsUser
Val=Validator,Eligibility,rtsUser

```

3. In `users.properties`, declare the user and password combinations.

For example:

```

rtsUser1=rtsUser1
rtsConfig=rtsConfig
rtsAdmin=rtsAdmin
Eli=Eli
Val=Val

```

What to do next

After you deploy the archive in the next step, you must upload all groups to the database from the Installation Settings wizard of the Decision Center console. For more information, see “Step 3: Set up groups” on page 45.

Declaring custom groups

If you create custom groups, you must declare them before you deploy the EAR file.

About this task

The Decision Center EAR file references the basic groups: `rtsUser`, `rtsConfigManager`, `rtsAdministrator`, and `rtsInstaller`.

You must add any custom groups that you declared in “Defining groups and permissions” on page 39. Also, add the `Validator` and `Eligibility` groups that you created for the Decision Center tutorials.

Tip:

- The EAR file is compressed. You must open it to extract the files that must be changed, and then replace the files in the EAR file. You can use Ant commands to repackage the EAR file, as explained in “Repackaging the Decision Center archive” on page 56.
- Back up the EAR file before you modify it.
- To use the Decision Center permissions mechanism, you must upload groups to the database. For more information, see “Completing the configuration from the Decision Center Enterprise console” on page 42.

Procedure

1. Add each custom group as a role in the appropriate application.xml file.
<ODM_InstallDir>/teamserver/applicationservers/JBoss5/jrules-teamserver-JBOSS5.ear/META-INF For example:
...
<security-role>
 <role-name>my_custom_group</role-name>
</security-role>
...
2. If you have already deployed the archive, upload the custom groups to the database from the Installation Settings wizard of the Decision Center console. For more information, see “Step 3: Set up groups” on page 45.

What to do next

Otherwise, proceed to “Step 4: Deploying the Decision Center EAR on JBoss” and upload the groups when you complete the configuration.

Step 4: Deploying the Decision Center EAR on JBoss

To deploy the Decision Center EAR on JBoss, you must copy the EAR archive to the deployment directory of your application server.

Procedure

1. Locate the following file: <ODM_InstallDir>/teamserver/applicationservers/JBoss5/jrules-teamserver-JBOSS5.ear
2. Copy the file to the deploy folder:
<JBoss_HOME>/server/default/deploy

Important:

Deploying the Decision Center EAR sets the persistence locale. After you save a rule to the database, you cannot change the persistence locale. If you want to install Decision Center in a language other than English, take note of the instructions provided in “Step 6: Completing your Decision Center configuration” on page 42.

Step 5: Verifying the deployment of the Decision Center consoles

After you have finished configuring Decision Center for your application server, verify that you have deployed the archive successfully.

About this task

You start your application server and then use your web browser to open the Decision Center Enterprise console.

Procedure

1. Make sure that your application server is running.
2. Start a new browser instance and enter the default URL to access Decision Center in a web browser.
 - The default URL to access Decision Center is `http://localhost:<PORT_NUMBER>/teamserver`. Set `<PORT_NUMBER>` to the port number of your web application. The default value of the port number is 8080. If your browser is not running on the same host as the application server, replace `localhost` with the address of the host.
 - Starting from WebSphere Application Server V8.0, each console has a different default URL.
 - For the Enterprise Console: `http://localhost:<PORT_NUMBER>/teamserver`
 - For the Business Console: `http://localhost:<PORT_NUMBER>/decisioncenter`
 - If your browser is not running on the same host as the application server, replace `localhost` with the address of the host.
 - If your web application is mapped to a host on a port that is different from the port number shown, change the port number to your host port number.The Decision Center log in page opens in your browser.
3. Sign in with `rtsAdministrator` rights to start testing.
For example, `rtsAdmin` and `rtsAdmin`.

Step 6: Completing your Decision Center configuration

After you have created a data source, defined security settings, and deployed the Decision Center EAR file, you complete the configuration either from the Decision Center console or by running Ant tasks. If your database is DB2 for z/OS®, generate the schema on z/OS for reasons of performance. You can, however, use the wizard to import extension files.

Completing the configuration from the Decision Center Enterprise console

After you have deployed the Decision Center archive to your application server, you can work from the Decision Center Enterprise console to complete or modify the configuration.

Installation Settings wizard overview:

You use the Installation Settings wizard in the Decision Center console to create or modify the database schema, set up message files or groups, or change the persistence locale or configuration parameters.

The Installation Settings wizard opens automatically when you start the Decision Center console to complete an installation.

You can also open the Installation Settings wizard by clicking **Configure > Installation Settings Wizard** in the Decision Center console after you have completed your initial installation. If you open Decision Center after you have

followed the steps to install the module, only the **Install** tab is available. For more information, see *Opening the Decision Center consoles*.

Note: To access the Installation Settings wizard, you must have both administrator privileges and the `rtsInstaller` role when you sign in.

You use the Installation Settings wizard for the following actions.

Table 2. Actions in the Installation Settings wizard

| Action | Description |
|---------------------------------|--|
| Configure the database | Mandatory when you complete the configuration with a database on a distributed platform. For more information, see Step 1: Configure the database. |
| Set up message files | Mandatory during the installation only if you have some custom rule model extension files. For more information, see Step 2: Set up message files. |
| Set up groups | You must set up the same groups that are declared in the application server if you want to use the Decision Center security and permissions mechanisms. For more information, see Step 3: Set up groups. |
| Change the persistence locale | Mandatory if the persistence locale is different from the locale <code>en_US</code> . For more information, see Step 4: Set the persistence locale. |
| Change configuration parameters | Optional. You change some configuration parameters when you customize Decision Center. For more information, see Step 5: Set configuration parameters. |

After you have completed the installation, Decision Center is ready to use but does not contain rule projects. If you open Decision Center and no rule projects are available, a message in the **Configure** tab informs you that no project has been found and that you should either publish a rule project by using Rule Designer or contact the administrator.

If you see this message, you must publish a rule project from Rule Designer. For more information, see *Publishing a project to Decision Center*.

More information about using the Installation Settings wizard is available from the Decision Center console online help. To access the online help, click **Help** in the top banner after you have signed in to Decision Center.

Note: If you have rule projects that were created with a previous product version, refer to the *Migrating topics* on how to upgrade the Decision Center database schema.

Step 1: Configure the database:

You use the Installation Settings wizard to configure the database.

About this task

You store the extensions to the Decision Center rule model in two XML files:

- Model description: This file usually has the file name extension `.brmx`.
- Initialization of enumerations and hierarchies: This file usually has file name extension `.brdx`.

For more information about defining common model extensions, see the customization topics.

Procedure

1. When the Installation Settings wizard opens in Decision Center, click **Next**.
2. Select one of the extension files.
 - **Default extensions** (already selected)
 - **Custom extensions (brmx/brdx)**, or
 - **Custom extensions (Zip)**
3. Click **Generate SQL** to generate the script that creates the database tables, which are based on the contents of your rule model files.
4. After the script is generated, select the **Execute the SQL script** check box, and then click **Next**.

Step 2: Set up message files:

Message files contain the display text that is associated with the extensions to the rule model contained in the .brmx and .brdx files.

About this task

You can find the default message file in `<ODM_InstallDir>/teamserver/bin/defaulttextextensionmessages_<LOCALE>.properties`.

If you use the default rule model when you create your database, the default message file is automatically sent to the database. To upload your own message files, use the Installation Settings wizard as explained below.

You must have a message file for each locale that you use. Message files are identified by their locale. The contents of the message files must respect the ISO-LATIN-1 standard.

Procedure

To declare a message file in the Installation Settings wizard:

1. Click **New**.
2. Enter a locale.
3. Browse to the location of the message file for this locale.
4. Click **Apply**.

Results

If Decision Center supports this locale, the Installation Settings wizard assigns a locale code so that you can identify it.

Example

For example:

```
status=Status
effectiveDate=Effective Date
expirationDate=Expiration Date
new=New
defined=Defined
```


Step 3: Set up groups:

In addition to creating groups in your application server when you set up security access, you must use the Setup Groups page in the Installation Settings wizard to upload groups to the database.

Before you begin

Before you set up groups in the Enterprise console, make sure to add all the groups that you want to see in the available list when enforcing project security or setting permissions in Decision Center. For more information, refer to the topics on Groups and Permissions in the Decision Center help.

About this task

You set up groups only if you want to use the Decision Center project access and permission mechanisms.

Tip: In Decision Center, the groups are the roles in the application server, **not** the groups defined in the user registry. Decision Center uses the group information to verify whether a user belongs to a role in the application server.

You do not have to upload the `rtsAdministrator` or `rtsInstaller` group. The administrator group has access to everything, and an installer user must belong to another group.

You use the Setup Groups page in the Installation Settings wizard to upload the default groups for `rtsUser` and `rtsConfigManager`, and any custom groups, such as `Validator` and `Eligibility` if you want to follow the permissions tutorial.

Procedure

To set up groups:

1. Click **New**.
2. Type the group name.
3. Click **Apply**.
4. Repeat steps 1 to 3 for each group.
5. When you have added all the groups, proceed in one of the following ways:
 - Click **Next** if you want to set a different persistence locale, or configuration parameters.
 - Click **Finish** if you do not want to change these settings.

Step 4: Set the persistence locale:

The persistence locale determines the language in which you store rules in the Decision Center database.

About this task

You set the locale when you deploy the Decision Center EAR file to your application server. As a consequence, you store the rules in the database in the locale of the Decision Center application.

Changing the persistence locale does not change the language in which Decision Center displays rules. Changing it in Decision Center is necessary only to match the locale of Rule Designer when you synchronize your rule projects, and to access the tutorials in your locale.

Important: You must not change the persistence locale after you have saved a rule to the database.

Procedure

To set the persistence locale:

1. Enter a locale in the **Locale** field.
2. Click **Apply**.
3. Proceed as follows:
 - Click **Next** if you want to set the configuration parameters.
 - Click **Finish** if you do not want to change these settings.

Step 5: Set configuration parameters:

Many tasks that are related to customizing Decision Center require that you add or remove configuration parameters.

About this task

Decision Center uses the following configuration parameters to generate complete URLs in permalinks:

- **teamserver.server.port:** the port number
- **teamserver.server.isSecure:** true if the connection is secure
- **teamserver.server.hostname:** the name of the host.

To create, modify, or delete configuration parameters, you use the Set configuration parameters page in the Installation Manager wizard. You generate these parameters when you sign in to the Decision Center console for the first time after you have configured the database. You can use the Installation Settings wizard to change these parameters at any time.

The following table gives a description of the main configuration parameters available in `teamserver.war/WEB-INF/lib/teamserver-model-XXX.jar/ilog/rules/teamserver/preferences.properties`.

Note:

The parameters in the table include the **teamserver** prefix, which is not in the `preferences.properties` file. You must include the prefix when you set configuration parameters in the Installation Settings wizard.

| Parameter | Used to |
|--|---|
| <code>teamserver.<extractorValidator>.class</code> | Specify a class of ruleset extractor validators to use for the <code>extractorValidator</code> name. The class must implement the <code>IlrExtractorValidator</code> interface. After you define this class, specify this name as the extractor validator to use when defining a ruleset extractor. |

| Parameter | Used to |
|---|---|
| <code>teamserver.build.path</code> | Define the location of the IRL cache in the file system. Compute the path as follows: <ul style="list-style-type: none"> Use this property with the name of the user who started the server as the root for the cache (<code><build.path>_<username></code>). If this property is not defined, use the system property <code>java.io.tmpdir</code> and add <code>rtscache</code>. For example, <code><temp dir>/rtscache_<username></code>. If the system property is not defined, use the server directory and add <code>rtscache</code>. For example, <code><server dir>/rtscache_<username></code>. |
| <code>teamserver.br1.verbalizers</code> | Specify the list of locales for which a BAL verbalizer is defined. |
| <code>teamserver.br1.verbalizer.<locale></code> | Specify the verbalizer class for the locale. The class must implement the <code>ilog.rules.vocabulary.verbalization.IlrVerbalizer</code> interface. |

Procedure

- Create a parameter or change an existing one.
 - To create a parameter, click **New**.
 - To change a parameter, select the check box next to the parameter and then click **Modify** to change the parameter or click **Delete** to remove it.
- Click **Apply** to implement your changes.
- Proceed as follows:
 - Click **Previous** if you want to make changes to previous settings.
 - Click **Finish**. The Installation log opens with a summary of the operations that you performed in the Installation Settings wizard.
- Click **OK** to finish.

What to do next

You now have to sign in to the Decision Center console. Continue with the section [Publishing a project](#).

Completing the configuration by using Ant tasks

Ant tasks provide an alternative method for completing or modifying the configuration. These tasks perform the same configuration steps as the Installation Settings wizard in the Decision Center console.

Setting up the Ant tasks environment:

To run Decision Center Ant tasks, you must first set up the required environment variables.

To run Ant tasks, you must have version 1.7.1 (or later) of Ant set up on your system. If Ant is not installed or your version is older than version 1.7.1, you must set up your environment to use the correct version of Ant. You can download Ant from the Apache web site, or you can use the Ant 1.7.1 distribution packaged at `<ODM_InstallDir>/shared/tools/ant`, where `<ODM_InstallDir>` is your Operational Decision Manager installation directory.

To test your current version of Ant:

Type the following command in a Windows command prompt or UNIX shell: `ant -version`

To set up your environment to use Ant:

1. Set the `ANT_HOME` environment variable to `<ODM_InstallDir>/shared/tools/ant`.
2. Set the `JAVA_HOME` environment variable to the path to your JDK installation (1.6).
3. Add the directory `<ODM_InstallDir>/shared/tools/ant/bin` to your `PATH` environment variable.

The Decision Center Ant tasks are defined in `<ODM_InstallDir>/teamserver/bin/build.xml` and executed by commands of the form:

```
ant <taskName> <parameters list>
```

Note: To execute these Ant tasks, you must use the same Java virtual machine version and vendor as the one that is used by the application server.

Ant task parameters start with `-D`. Use them to set values such as the following ones:

- `-Dserver.url=<server url>`: Specifies the URL of the target application server.
- `-DdatasourceName=<data source name>`: Specifies the JNDI name of the data source to use for the task. The default value is `jdbc/ilogDataSource`.

```
ant execute-schema -Dserver.url=<protocol://host:port>/teamserver/  
-DdatasourceName=jdbc/ilogDataSource -Dfile=my_sql_file.sql
```

The `<protocol://host:port>` URL is defined in the file `<ODM_InstallDir>/teamserver/bin/teamserver-anttasks.properties`. If your browser is not running on the same host as the application server, replace `localhost` with the address of the machine. If your web application is mapped to a host on a port that is different from the port number shown, change the port number to your host port number.

The file `<ODM_InstallDir>/teamserver/bin/teamserver-anttasks.properties` defines the value of some common parameters and others that depend on the application server used. You do not have to include these parameters in your Ant task command if they are properly defined in this file. The content of the `teamserver-anttasks.properties` file is as follows:

```
# Default properties  
# -----  
rtsAdmin.login=rtsAdmin  
rtsAdmin.password=rtsAdmin  
  
protocol=http  
server.host=localhost  
server.port=8080  
server.url=${protocol}://${server.host}:${server.port}/teamserver  
  
datasourceName=jdbc/ilogDataSource  
  
outputFile=output.sql  
  
languagePackPath = .  
languagePackOutputPath = ./generated
```

```
persistenceLocale =
selector =
branch =
override = false
```

Also, take note of any special instructions in this file concerning your application server.

The **appserver.name** property configures the class path for the Ant tasks. If you have to add specific drivers to your class path, you can add them to `<ODM_InstallDir>/teamserver/lib/classpath-teamserver.xml`.

Communication between the Ant tasks and Decision Center supports the HTTP or HTTPS communication protocols. For more information, see Ant task communication protocol.

Creating the database schema:

You can create the database schema in a single operation by using the **set-extensions** Ant task, or choose to create it step by step.

Creating schemas with the set-extensions Ant task:

For convenience, you can create the database schema by using the **set-extensions** Ant task.

About this task

Extensions to the Decision Center rule model are stored in two XML files.

- Model description: This file usually has the file name extension `.brmx`.
- Initialization of enumerations and hierarchies: This file usually has file name extension `.brdx`.

You can use Ant tasks to load the rule model from the two XML files and build the SQL script that is necessary to get the proper database schema.

Note: To run these Ant tasks, you must use the same Java virtual machine version and vendor as the one used by the application server.

Alternatively, you can create the database schema step by step, which is useful if you want to look at the generated SQL schema. See “Creating the schema using a step-by-step sequence” on page 50.

Procedure

Run the **set-extensions** Ant task.

This Ant target runs **gen-create-schema** + **execute-schema** + **upload-extensions** + **upload-roles**. Set the following parameters:

-Dserver.url=<server url>

-DdatasourceName=<data source name>

-DextensionModel=<model file>

The model description file, with the `.brmx` extension

-DextensionData=<data file>

The model data description, with `.brdx` extension

[`-DdbName=<database schema name>`]

You can use this optional parameter to specify the database schema name. Otherwise, Decision Center uses the database user name as the schema name. However, some databases allow some users to access several schemas, and the default schema does not always reflect the user name.

[`-Droles=<role list>`]

You can use this optional parameter to upload the list of roles to Decision Center. This list is specified as "role1 role2". For example:

```
ant upload-roles -Droles="rtsUser rtsConfigManager Eligibility Validator"
```

Creating the schema using a step-by-step sequence:

If you want to look at the generated SQL schema, you can create it step-by-step.

Creating the database schema script:

You can create the database schema script by using the **gen-create-schema** Ant task.

Procedure

To create the SQL script that is necessary to create or update the database schema, run the **gen-create-schema** Ant task with these parameters:

`-Dserver.url=<server url>`

`-DdataSourceName=<data source name>`

`-DextensionModel=<model file>`

The model description (.brmx extension).

`-DextensionData=<data file>`

The model data description (.brdx extension).

[`-DdbName=<database schema name>`]

You can use this optional parameter to specify the database schema name in which the Decision Center tables are stored. If you do not specify the parameter, Decision Center uses the database user name as the schema name. However, some databases allow some users to access several schemas, and the default schema is not always named the same as the user.

[`-DoutputFile=<SQL file>`]

The name of the file that stores the generated SQL script. If this parameter is not specified, the task creates a file named `output.sql` in the directory that is defined as `basedir` in the `build.xml` file.

```
ant gen-create-schema -DextensionModel=my_model_file.brmx -DextensionData=my_data_file.brdx -DoutputFile=output.sql
```

Results

1. The task connects to the specified data source from the application server.
2. The task checks whether this data source points to an existing Decision Center database.
 - If a database does not exist, the task builds the SQL script to create a fresh database schema to store the model.
 - If a database does exist, the task builds the SQL script that is necessary to update the existing database schema.

Executing the database schema script:

You execute the database schema script.

Procedure

To execute the SQL script that you created, run the **execute-schema** Ant task with these parameters:

-Dserver.url=<server url>

-DdatasourceName=<data source name>

[-Dfile=<SQL file>]

The name of the file to execute, which corresponds to the script that you created. If you do not specify this parameter, the task attempts to execute a file named `output.sql` in the directory that is defined as `basedir` in the `build.xml` file.

```
ant execute-schema -Dfile=my_sql_file.sql
```

Uploading the database schema extension:

You upload the database schema extension.

Procedure

To store the rule model description in the database schema, run the **upload-extensions** Ant task with these parameters:

-Dserver.url=<server url>

-DdatasourceName=<data source name>

-DextensionModel=<model file>

The model description (`.brmx` extension).

-DextensionData=<data file>

The model data description (`.brdx` extension).

```
ant upload-extensions -DextensionModel=my_model_file.brmx -DextensionData=my_data_file.brdx
```

Results

The description is stored in the database so that Decision Center applications can load it when they start. It is also used by **gen-create-schema** to get the current model description to run a `diff` with the new schema.

In a cluster, you must restart the servers and close all current sessions.

Uploading a list of roles or groups to the database:

In addition to creating groups in your application server when you set up security access, you must upload groups to the database.

Before you begin

1. Add all the groups that you want to see in the available list when you enforce project security or setting permissions in Decision Center.
2. Create the default groups for `rtsUser` and `rtsConfigManager`, and upload your custom groups.

About this task

You must upload roles or groups only if you want to use the Decision Center project access and permissions mechanisms. For more information, see the topics on Groups and Permissions in the Decision Center online help.

You do not have to upload the `rtsAdministrator` group or the `rtsInstaller` group. The Administrator group has access to everything and an Installer user must belong to another group.

Note: To run these Ant tasks, you must use the same Java virtual machine version and vendor as the one used by the application server.

Procedure

To store in the database the list of roles or groups to be used by the application, run the **upload-roles** Ant task with the following parameters:

-Dserver.url=<server url>

-DdatasourceName=<data source name>

-Droles=<role list>

<role list> is the list of roles or groups to upload to Decision Center, specified as "group1 group2".

```
ant upload-roles -Droles="rtsUser rtsConfigManager Eligibility Validator"
```

Removing a database schema:

You can create an SQL script to remove (drop) a database schema by using the **gen-drop-schema** Ant task.

About this task

To remove a database schema, you proceed in two steps:

1. Create the SQL script that is necessary to remove the database schema.
2. Execute the SQL script that you created.

Procedure

1. To create the SQL script to delete a database schema, run the **gen-drop-schema** Ant task with the following parameters:

-Dserver.url=<server url>

-DdatasourceName=<data source name>

-DextensionModel=<model file>

The description of the database schema to remove.

[-DdbSchemaName=<database schema name>]

You can use an optional parameter to specify the database schema name. If you do not specify this parameter, Decision Center uses the database user name as the schema name. However, in some databases, users can access several schemas and the default schema is not always named as the user.

[-DoutputFile=<SQL file>]

The name of the file that stores the generated SQL script. If you do not specify this parameter, the task creates a file named `output.sql` in the directory that is defined as `basedir` in the `build.xml` file.

- ```
ant gen-drop-schema -DextensionModel=my_model_file.brmx -DoutputFile=my_sql_file.sql
```
2. To execute the SQL script that you created, run the **execute-schema** Ant task with these parameters:

```
-Dserver.url=<server url>
-DdataSourceName=<data source name>
[-Dfile=<SQL file>]
```

The name of the file to execute, which corresponds to the script that you created. If you do not specify this parameter, the task attempts to execute a file named `output.sql` in the directory that is defined as `basedir` in the `build.xml` file.

```
ant execute-schema -Dfile=my_sql_file.sql
```

The task connects to the specified data source from the application server. It reads the model description that is passed in the parameters, and generates the SQL script to remove the existing schema. Because many database tables are linked through foreign keys, these tables must be removed in a specific order and the script generation handles these constraints.

### Example

Here is the complete code sample:

```
ant gen-drop-schema -DextensionModel=my_model_file.brmx -DoutputFile=my_sql_file.sql
ant execute-schema -Dfile=my_sql_file.sql
```

### Defining and uploading message files:

You can define and upload message files to Decision Center by using the **upload-messages** Ant task.

Message files contain the display text that is associated with the extensions to the rule model that is contained in the `.brmx` and `.brdx` files. For example:

```
status=Status
effectiveDate=Effective Date
expirationDate=Expiration Date
new=New
defined=Defined
```

The default messages file is provided in: `<ODM_InstallDir>/teamserver/bin/defaulttextensionmessages_<LOCALE>.properties`

**Note:** The contents of the messages files must conform to the ISO-LATIN-1 standard.

You must have a messages file for each locale that you use. Upload the messages file to Decision Center by running the **upload-messages** Ant task with these parameters:

- **-Dserver.url=<server url>**
- **-DdataSourceName=<data source name>**
- **-Dlocale=<locale>**
- **-DmessageFile=<message file>**

```
ant upload-messages -Dlocale=en_US -DmessageFile=mymessages.properties
```

## Setting the persistence locale:

The persistence locale is used to determine the language in which rules are stored in the Decision Center database.

### About this task

The persistence locale is set when you deploy the Decision Center archive to your application server, which means that the rules in the database are stored in the locale of the Decision Center application.

Changing the persistence locale does not change the language in which rules display in Decision Center. Changing the persistence locale in Decision Center is necessary only to match the locale of Rule Designer when you synchronize your rule projects, and to access the tutorials in your locale.

**Important:** You must not change the persistence locale after you have saved a rule to the database.

### Procedure

To set the persistence locale by running an Ant task:

1. Open the `<ODM_InstallDir>/teamserver/bin/teamserver-anttasks.properties` file.

This file defines the value of some common parameters.

2. Add your locale to the **persistenceLocale** property and save the `teamserver-anttasks.properties` file.

For example: `persistenceLocale = fr_FR`

3. Run the Ant task in this form: `ant taskName parameters_list`

Alternatively, you can add the parameter to the command line. For example:

```
ant taskName -DpersistenceLocale=fr_FR
```

### Adding or removing configuration parameters:

For many tasks that are related to customizing Decision Center, you must add or remove configuration parameters.

The following configuration parameters, used to generate complete URLs in permalinks, are generated the first time you sign in to Decision Center after the database is configured. You can use the Installation Settings wizard to set these parameters beforehand or change them afterward:

- **teamserver.server.port:** The port number
- **teamserver.server.isSecure:** true if the connection is secure.
- **teamserver.server.hostname:** The name of the host

The following table gives a description of the main configuration parameters available in `teamserver.war/WEB-INF/lib/teamserver-model-XXX.jar/ilog/rules/teamserver/preferences.properties`.

| Parameter                                                | Use                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|----------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>teamserver.&lt;extractorValidator&gt;.class</code> | Specify a ruleset extractor validator class to use for the <code>extractorValidator</code> name. The class must implement the <code>ILrExtractorValidator</code> interface. After this class is defined, specify this name as the extractor validator to use when defining a ruleset extractor.                                                                                                                                                                                                                                                                                                                                                                                                           |
| <code>teamserver.build.path</code>                       | Define where the cache of the IRL is located on the file system. The path is computed as follows: <ol style="list-style-type: none"> <li>1. First, use this property with the name of the user who started the server as the root for the cache:<br/><code>&lt;build.path&gt;_&lt;username&gt;</code>.</li> <li>2. If the path is not defined, use the system property <code>java.io.tmpdir</code> and add <code>rtscache</code>. For example,<br/><code>&lt;temp_dir&gt;/rtscache_&lt;username&gt;</code>.</li> <li>3. If the system property is not defined, use the server directory and add <code>rtscache</code>. For example, <code>&lt;server_dir&gt;/rtscache_&lt;username&gt;</code>.</li> </ol> |
| <code>teamserver.br1.verbalizers</code>                  | Specify the list of locales for which a BAL verbalizer is defined.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <code>teamserver.br1.verbalizer.&lt;locale&gt;</code>    | Specify the verbalizer class for the specified locale. The class must implement the <code>ILrVerbalizer</code> interface.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |

You can use the following Ant tasks to add or remove configuration parameters:

#### set-config-param

Sets a configuration parameter for a specified user. If the user is not specified, it sets a *global parameter*.

Parameters:

- `-Dserver.url=<server url>`
- `-DdatasourceName=<data source name>`
- `[-Duser=<username>]`
- `-Dkey=<parameter key>`
- `-Dvalue=<parameter value>`

For example:

```
ant set-config-param -Dkey=locale -Dvalue=en_US
```

#### remove-config-param

Drops the configuration parameter for a specified user. If the user is not specified, it drops the global configuration parameter.

Parameters:

- `-Dserver.url=<server url>`
- `-DdatasourceName=<data source name>`
- `[-Duser=<username>]`
- `-Dkey=<parameter key>`

## print-config-param

Prints the global parameters or specified user parameters if the *username* value is specified. If no key is specified, all keys are printed.

Parameters:

- **-Dserver.url=<server url>**
- **-DdataSourceName=<data source name>**
- **[-Duser=<username>]**
- **-Dkey=<parameter key>**

## Repackaging the Decision Center archive:

You can repack the Decision Center archive by using an Ant task.

When you add new .jar files to the Decision Center archive, you must repack the archive by running the **repackage-ear** or **repackage-war** Ant task. This task does not use the **server.url** and **dataSourceName** parameters.

If you have customized Decision Center, you must package the custom .jar files before you use the Ant task to repack the Decision Center.

The **repackage-ear** or **repackage-war** Ant task takes the following parameters:

**-DtargetEar=<target ear>**

**-DtargetWar=<target war>** for the Tomcat application server

**-DsourceEar=<source ear>**

**DsourceWar=<source war>** for the Tomcat application server

**-DdescriptorsDir=<descriptors directory>**

A directory that is copied into the META-INF directory of the target EAR (not mandatory).

**-DadditionalJars=<"myjar1.jar,myjar2.jar, myjarn.jar">**

Additional .jar files to store in the lib directory of the target archive (not mandatory).

**-DtmpDir=<directory>**

A directory that you can specify to store temporary files (not mandatory).

**-DwebResourcesDir=<web resources directory>**

A directory that is copied into the WAR library (not mandatory).

**-Dconsole=both|enterprise|business**

Specifies whether to repack the Business or the Enterprise WAR files. The default is both.

## Configuring the search function of the Decision Center Business console

The search function in the Decision Center Business console is based on the Solr search engine. You must configure the engine to provide this functionality.

### Using a remote Solr search engine

You can set up the search function in Decision Center Business console to work with a remote instance of the Apache Solr search engine.

## About this task

The search function in Decision Center Business console uses an embedded instance of the Apache Solr search engine. Alternatively, you can have the search function work with a remote instance of the search engine, which you can run on another computer, or the same computer but in a dedicated web application.

## Procedure

To configure the search function to run with a remote instance of the Solr search engine:

1. Install the Apache Solr search engine on another computer, or as part of a dedicated web application on your computer.

For information on installing the Solr server, visit the Apache Solr website.

2. Locate the `decisioncenter-solr-home.zip` file in the Decision Center `teamserver` folder on your computer.

The Decision Center installation program placed the folder on your computer.

3. Decompress the file in a directory in the remote instance of the Solr server.
4. Configure the home directory of the remote Solr server to use the location of the decompressed configuration files.

For more information, see the documentation on the Apache Solr website.

5. Configure the Decision Center `preferences.properties` file to point to the URL of the Solr server.

## Setting parameters for the Solr search engine

You can configure the search function in the Decision Center Business console to run with the Apache Solr search engine.

Decision Center Business console provides a function for searching rule projects. When you install this feature, it uses an embedded instance of the Apache Solr search engine. Alternatively, you can have the search function run with a remote instance of the search engine, which you can install on another computer, or on the same computer but in a dedicated web application. For more information, see “Using a remote Solr search engine” on page 56.

You can set three parameters for the search engine:

*Table 3. Search parameters*

| Parameter                               | Description                                                                                                                                                                                                                                                                                       |
|-----------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>search.SearchProvider</code>      | This parameter takes one of the following values: <ul style="list-style-type: none"><li>• <code>SolrEmbedded</code>: Use this value to select the embedded Solr search engine.</li><li>• <code>SolrRemote</code>: Use this value to select a remote instance of the Solr search engine.</li></ul> |
| <code>search.SolrEmbeddedDataDir</code> | Use this optional parameter to direct the index of the embedded version ( <code>SolrEmbedded</code> ) to a specific directory on the Decision Center.                                                                                                                                             |
| <code>search.SolrRemoteUrl</code>       | Use this parameter with <code>SolrRemote</code> to provide the URL of the remote Solr search engine.                                                                                                                                                                                              |

You set the configuration parameters for the search engine within the `preferences.properties` file for Decision Center. For more information on the `preferences.properties` file, see Configuration parameter maintenance.

The following table provides examples for setting the parameters in `preferences.properties` to work with the Solr search engine:

Table 4. Configuration table for search properties

| Solr server                                                                                                                                                                                          | Parameter settings                                                                                                                                                   |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>External server</p> <p>The URL of the external server depends on the installation. For this example, the remote address of the search engine is <code>http://mysearchserver:8983/solr</code>.</p> | <p>Set the <code>preferences.properties</code> file as follows:</p> <pre>search.SearchProvider=SolrRemote search.SolrRemoteUrl=http://mysearchserver:8983/solr</pre> |
| <p>Embedded server</p> <p>In this example, you store the index in the <code>c:/temp/DC-SearchIndex</code> directory on your computer.</p>                                                            | <p>Set the <code>preferences.properties</code> file as follows:</p> <pre>search.SearchProvider=SolrEmbedded search.SolrEmbeddedDataDir=c:/temp/DC-SearchIndex</pre>  |

If you do not specify a directory for `search.SolrEmbeddedDataDir`, Decision Center stores the search index in a temporary directory. When the server stops running, it also stops using the temporary directory. When the server restarts, it creates a new temporary directory, and completely re-indexes the repository.

If you specify a directory for `search.SolrEmbeddedDataDir`, the directory and its content persist across server restarts, and the server does not re-index the repository with each restart.

## Additional steps to configure Decision Validation Services

There are additional steps that you must perform to configure Decision Validation Services.

### Before you start

Before deploying and configuring Decision Validation Services, you must complete some steps.

1. Install Rule Execution Server.
2. Install Decision Center.
3. Configure both of these applications on the same application server.

After these installations are complete, see “What steps to follow” to understand the additional tasks that you might have to complete.

### Note:

Only users belonging to the `resAdministrators` or `resMonitors` groups are able to see the Decision Warehouse tab in the Rule Execution Server console after Decision Validation Services is configured.

### What steps to follow

The configuration steps you follow to configure Decision Validation Services are summarized in the following table:

| Step                                                                                                                                                                                                                                                                 | Required                                                                                                                                                                                                                                                                                                             |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| "Step 1: Creating Decision Warehouse database resources"                                                                                                                                                                                                             | Optional. If you have used the Rule Execution Server console to create database resources, you do not have to perform this task. See Creating a database schema by using the Rule Execution Server console for more information.                                                                                     |
| <p data-bbox="459 489 1125 552">"Step 2: Deploying the Decision Validation Services archive for the first time" on page 61</p> <p data-bbox="459 573 1125 636">This archive is the default SSP archive packaged with the Decision Validation Services installer.</p> | <p data-bbox="1125 489 1455 573">Optional. This steps allows you to check the availability of the feature.</p> <p data-bbox="1125 573 1455 825"><b>Note:</b><br/>You can also use the default SSP archive if you have an XML XOM. In this case, you do not have to repackage the archive to include the XML XOM.</p> |
| "Step 3: Checking the availability of Decision Validation Services" on page 61                                                                                                                                                                                       | Optional.                                                                                                                                                                                                                                                                                                            |
| "Step 4: Packaging Decision Validation Services by using Ant" on page 62                                                                                                                                                                                             | Optional. You can use this task as an alternative method for configuring or modifying the Decision Validation Services installation.                                                                                                                                                                                 |

## Step 1: Creating Decision Warehouse database resources

You can use SQL scripts to create a dedicated schema in the database.

If you set the Rule Execution Server persistence to `datasource` or `jdbc` and you intend to use Decision Warehouse to store your test execution results, you must create a dedicated schema in the database that contains these tables and views. To do so, you can use the provided SQL scripts, located in `<ODM_InstallDir>/executionserver/databases`. A readme file in this directory provides additional information about the scripts.

The script that creates the Decision Warehouse database schema is named `trace_<database_name>.sql`.

### Note:

The Installation Settings wizard in the Rule Execution Server console creates all the necessary tables for Rule Execution Server and for Decision Warehouse. If you are configuring Decision Validation Services and you have already run the Installation Settings wizard to create the tables, you do not have to create database resources manually. However, if you did not use the wizard to create database resources, you must run the script to create the Decision Warehouse database schema.

When you use DB2 (except on z/OS), the scripts that create the Rule Execution Server database tables are written for databases that use automatic storage.

- BP32K is the buffer pool that is expected in SYSCAT.BUFFERPOOLS. If BP32K is not there, you can use the existing buffer pool or create a new buffer pool named BP32K. Use the following command to query SYSCAT.BUFFERPOOLS for the existing buffer pool:

```
Select * from SYSCAT.BUFFERPOOLS
```

Otherwise, use the following command to create a buffer pool named BP32K:

```
CREATE BUFFERPOOL BP32K SIZE 2000 PAGESIZE 32K
```

- You must update the trace\_db2.sql script and select the custom option in the Installation Settings wizard to run it. Modify the following line in the script to specify storage for the table space:

```
CREATE TABLESPACE RESDWTS PAGESIZE 32K BUFFERPOOL BP32K;
```

Here is an example of the table space specification in the script:

```
CREATE TABLESPACE RESDWTS PAGESIZE 32K MANAGED BY Database
USING [FILE 'C:\DB2\Container.file' 640] BUFFERPOOL BP32K;
```

- You might have to further modify the script based on your database settings.

**Note:**

If you completed this task as part of the Rule Execution Server configuration, you do not have to do this a second time. It is included here solely for completeness.

If you use Command Editor to run the scripts, you must log in with the credentials that you use for the data source for Rule Execution Server.

Use any tool that can handle SQL to import and run the SQL scripts. The tools provided for each database include:

| Database    | Database tool                  |
|-------------|--------------------------------|
| IBM DB2     | DB2 command line processor     |
| Derby       | ij command line processor      |
| MySQL       | mysql command line processor   |
| Oracle      | sqlplus command line processor |
| Postgre SQL | Postgre SQL command line tool  |
| SQL Server  | Query Tool                     |
| Sybase      | isql command line processor    |

To access the database, the database user must have the following credentials:

- A user ID and a password
- Complete privileges on the tables and view of the schema (create, insert, delete)
- create index privileges
- On Oracle, create trigger and create sequence privileges. If you use an Oracle database, run all the scripts in the SQL Plus client.

Install a database client for the database that you use. Refer to the documentation of the database for more information.

The default CLOB size might not be sufficient for the FULL\_EXECUTION\_TRACE field in the EXECUTION\_TRACES table. You might need a size qualifier if SQL raises exceptions with the <Lob-Value> reason code.



## Step 2: Deploying the Decision Validation Services archive for the first time

Deploy the Decision Validation Services archive for the first time.

### About this task

If you want to check the availability of Decision Validation Services, you must deploy the default SSP (Scenario Service Provider) EAR as described below.

**Note:** You must deploy the archive on the same server as the execution unit (XU).

### Procedure

Copy the JBoss file `<ODM_InstallDir>/executionserver/applicationservers/JBoss5/jrules-ssp-JBOSS5.ear` to `<JBOSS_HOME>/server/<server configuration>/deploy`.

You can use this procedure to deploy the default SSP archive packaged with the installer or any subsequent deployment of a repackaged archive.

If you have an XML XOM, you can also use the SSP EAR to test your rules.

**Note:** Because the ruleset archive inside a RuleApp includes an XML XOM, you do not have to repackage the SSP EAR to include the XOM.

## Step 3: Checking the availability of Decision Validation Services

A specific testing URL is designed for you to check the rule session type, the DAO factory class for trace persistence, the class used to cache the Decision Validation Services job, and the pool size for asynchronous execution.

To check the availability of Decision Validation Services:

1. Enter the URL `http://<host>:<port>/testing` in a web browser.
2. Log in to the (Scenario Service Provider) SSP application in one of the Rule Execution Server roles.

The application displays a home page, which contains information about the SSP server.

### Version

The version of Decision Server used.

### Patch level

The patch level of Decision Server used.

### License information

The type of license of this version.

### RuleSession

The rule session type (POJO or J2SE).

### DAO Factory Class

The DAO (Data Access Object) factory class that is used to persist the trace into the data warehouse.

### Job store class

The name of the class that is used to persist the Decision Validation Services job into a cache to free the memory during long computations.

### Job pool size

The size of the pool for asynchronous execution.

#### Started since

The time and date when the SSP started.

#### Jobs currently running

The About screen provides information about the jobs that are currently running after you run Decision Validation Services in Decision Center:

- A **Job ID** is listed in the table when a user clicks **Run** in Decision Center.
- The **Created** column records the date and time when each job is initialized.
- The **Status** column shows the number of scenarios that have already been tested, compared to the total number of scenarios.
- The **Start time** records the time when a resource is allocated for the job.
- The **Parts** column records the number of parts in the job:
  - A job that is not executed in parallel has one part.
  - A job that is executed in parallel has one or more parts.
- The **End time** records the time when the execution of the job is complete, that is, all of the scenarios in the job have been tested.

The report for the job is automatically downloaded by Decision Center at the end of the execution. If the scenario suite is run in the background, the user downloads the report by viewing the list of scenario suites, and then clicking the report link when it becomes available. After the report is viewed, the job is removed from the table. The job remains in the table until the report is downloaded.

## Step 4: Packaging Decision Validation Services by using Ant

You can configure Decision Validation Services archives by using an Ant task.

### About this task

This section is for users who install the product on Windows and other supported distributed platforms only.

You can configure Decision Validation Services archives by running the **ssp-setup** Ant task.

### Procedure

1. Define the Ant task in your build file by using the `<taskdef>` Ant element in one of the following ways:
  - Define the task at the top level, or within a specific target.

```
<taskdef resource="res-tasks.properties"
classpath="${<InstallDir>}/executionserver/lib/jrules-res-setup.jar"/>
```
  - If the JAR file is available in your system, you can write the following code:

```
<taskdef resource="res-tasks.properties"/>
```
2. Use the **ssp-setup** Ant task to update a Scenario Service Provider (SSP) artifact to your specific configuration and XOM.
3. Run the Ant task in one of the following ways:
  - From the command line: Execute Ant in the appropriate directory, followed by the name of the build file if necessary.
  - From Eclipse: Right-click the Ant file and click **Run**.

## What to do next

You can now deploy the testing archive.

---

## Configuring the Decision Center consoles on JBoss 6.1 EAP

To use the Decision Center Enterprise console on a new instance of the JBoss application server, you must deploy the provided archive for this server and perform a number of configuration tasks.

### Before you start

Review the steps that you must perform to complete the configuration.

These instructions are intended for users familiar with JBoss and the database they are using. Examples using the Derby embedded database are included throughout to help beginners.

Refer to “What steps to follow” to understand the steps that you must follow to configure Decision Center on your application server.

You can complete the configuration of your application server with the Installation Settings wizard in the Decision Center console or with Ant scripts.





If you need to configure Decision Validation Services, see “Additional steps to configure Decision Validation Services” on page 85.



After finishing the configuration, you can use Decision Center (see Opening the Decision Center consoles). There is no rule project the first time you open the console. You must publish a project from Rule Designer (see Publishing a project to Decision Center).

**Note:** If you have rule projects created with a previous product version, refer to the Migrating topics on how to upgrade the Decision Center database schema.

### What steps to follow

The steps that you follow to configure Decision Center on JBoss 5 and 6.1 are summarized in the following table:

| Step                                                            | Required                                                                                                                              |
|-----------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------|
| “Step 1: Restricting database user permissions” on page 64      |                                                  |
| “Step 2: Creating a data source and connection pool” on page 65 |                                                  |
| “Step 3: Configuring security on JBoss” on page 66              | “Defining groups and permissions” on page 66<br> |
|                                                                 | “Declaring custom groups” on page 67<br>Optional                                                                                      |
| “Step 4: Deploying the Decision Center EAR on JBoss” on page 68 |                                                  |

| Step                                                                                                                                                                                                                                                                                                         | Required                                                                              |                                                                                       |  |                                                              |                                                                                                                                                                                          |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|--|--------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| “Step 5: Verifying the deployment of the Decision Center consoles” on page 68                                                                                                                                                                                                                                | Recommended                                                                           |                                                                                       |  |                                                              |                                                                                                                                                                                          |
| <table border="1"> <tr> <td>“Step 6: Completing your Decision Center configuration” on page 69</td> <td>“Completing the configuration from the Decision Center Enterprise console” on page 69</td> </tr> <tr> <td></td> <td>“Completing the configuration by using Ant tasks” on page 74</td> </tr> </table> | “Step 6: Completing your Decision Center configuration” on page 69                    | “Completing the configuration from the Decision Center Enterprise console” on page 69 |  | “Completing the configuration by using Ant tasks” on page 74 | <br>Use one of the two methods to complete the installation. Required for distributed platforms only. |
| “Step 6: Completing your Decision Center configuration” on page 69                                                                                                                                                                                                                                           | “Completing the configuration from the Decision Center Enterprise console” on page 69 |                                                                                       |  |                                                              |                                                                                                                                                                                          |
|                                                                                                                                                                                                                                                                                                              | “Completing the configuration by using Ant tasks” on page 74                          |                                                                                       |  |                                                              |                                                                                                                                                                                          |
| Verifying your configuration of Decision Center                                                                                                                                                                                                                                                              |    |                                                                                       |  |                                                              |                                                                                                                                                                                          |

## Step 1: Restricting database user permissions

Decision Center data is stored in a database. The database administrator might require that you provide the specific permissions that you need when accessing this database.

**Note:** This step applies when database access needs to be restricted. If you manage the database yourself (for example, you use an embedded database for test purposes) or if you do not need further restrictions, skip this step and proceed to the next configuration step.

Connection to the Decision Center database, established in the data source credentials, and any subsequent requests to the database are handled through a database user. This database user (name and password), for example `rtsdbUser`, is defined by the database administrator and has no relation to the standard Decision Center groups.

The following table gives the permissions that the database administrator must define on the Decision Center database, with attention given to the type of operations that you want to perform. Some supported databases do not require all the above permissions.

| Database permission | Operation           |                                                                              |                                                                              |                             |
|---------------------|---------------------|------------------------------------------------------------------------------|------------------------------------------------------------------------------|-----------------------------|
|                     | Use Decision Center | Create the database schema by using the Decision Center console or Ant tasks | Modify the database schema by using the Decision Center console or Ant tasks | Migrate the database schema |
| ALTER TABLE         | Not required        | Not required                                                                 | Required                                                                     | Required                    |
| CREATE INDEX        | Not required        | Required                                                                     | Required                                                                     | Required                    |
| CREATE ROLE         | Not required        | Not required                                                                 | Not required                                                                 | Required                    |
| CREATE SEQUENCE     | Not required        | Required                                                                     | Required                                                                     | Required                    |
| CREATE TABLE        | Not required        | Required                                                                     | Required                                                                     | Required                    |
| CREATE VIEW         | Not required        | Required                                                                     | Required                                                                     | Required                    |
| DROP INDEX          | Not required        | Not required                                                                 | Required                                                                     | Required                    |

| Database permission | Operation           |                                                                              |                                                                              |                             |
|---------------------|---------------------|------------------------------------------------------------------------------|------------------------------------------------------------------------------|-----------------------------|
|                     | Use Decision Center | Create the database schema by using the Decision Center console or Ant tasks | Modify the database schema by using the Decision Center console or Ant tasks | Migrate the database schema |
| DROP SEQUENCE       | Not required        | Not required                                                                 | Required                                                                     | Required                    |
| DROP TABLE          | Not required        | Not required                                                                 | Required                                                                     | Required                    |
| DROP VIEW           | Not required        | Not required                                                                 | Required                                                                     | Required                    |
| INSERT TABLE        | Required            | Required                                                                     | Required                                                                     | Required                    |
| SELECT SEQUENCE     | Required            | Required                                                                     | Required                                                                     | Required                    |
| SELECT TABLE        | Required            | Required                                                                     | Required                                                                     | Required                    |
| UPDATE TABLE        | Required            | Required                                                                     | Required                                                                     | Required                    |

## Step 2: Creating a data source and connection pool

You declare a data source to store the data that is used by Decision Center.

### Before you begin

You must have a database to create a data source.

**Note:** Beginners who use the Derby embedded database can create the database as part of this procedure.

For a list of supported databases, see the IBM Support site.

### Procedure

- Register the JDBC driver as a core module.
  - Under the `<JBoss_HOME>/modules` directory, create a file path structure. For example, for a Derby driver, create the following structure:  
`<JBoss_HOME>/modules/org/apache/derby/main`
  - Copy the JDBC driver `.jar` file into the main subdirectory.
  - In the main subdirectory, create a `module.xml` file. Example for Derby:

```
<module xmlns="urn:jboss:module:1.0" name="org.apache.derby">
 <resources>
 <resource-root path="derby.jar"/>
 <!-- Insert resources here -->
 </resources>
 <dependencies>
 <module name="javax.api"/>
 </dependencies>
</module>
```
  - Launch the JBoss server and run the `jboss-cli` command to register the new module in the server. The command is available in `<JBoss_HOME>/bin`. It must be executed with the name of the new module as follows:  
`./jboss-cli[.sh|.bat] --connect "/subsystem=datasources/jdbc-driver=derby:add(driver-name=derby,driver-module-name=org.apache.derby)"`
- Create the data source.

- a. Log in to the JBoss Management Console.
  - b. Click the **Profile** tab in the top right of the console.
  - c. From the tree on the left, go to **Subsystems > Connector > Datasources**
  - d. Click the **Datasources** panel.
  - e. Click **Add**, and then enter a name and a JNDI name (java:/jdbc/i1ogDataSource) as the data source attributes.
  - f. Select the JDBC driver to be used for the new data source. The JDBC driver registered as a core module in step 1 should appear here.
3. Specify the connection settings.

#### Connection URL

This setting defines the JDBC connection URL. For example:  
`jdbc:derby:c:/rtsdb;create=true`

#### User name and Password

The tags `username="rtsdbUser"` and `password="rtsdbUser"` are used to access the database.

**Note:** If the transaction isolation level is not set to `READ-COMMITTED`, Decision Center might encounter various database deadlocks. Therefore, you might have to configure data source isolation for better concurrency.

4. Enable the new data source. After the data source is created, it appears in the Available Datasources table. By default, the new data source is not enabled.
  - a. Select the new data source in the table and then click **Enable**.
  - b. Confirm the operation. The data source is listed as **Enabled**.

## Step 3: Configuring security on JBoss

You control access to Decision Center and enforce security by defining user groups and associated roles.

### Defining groups and permissions

You control access to Decision Center and enforce security by defining user groups and associated roles.

#### About this task

Any user of Decision Center must belong to at least one of these mandatory groups:

- `rtsAdministrator`
- `rtsConfigManager`
- `rtsInstaller`
- `rtsUser`

Adherence to these groups determines what parts of Decision Center a user can access. You must create all these groups. For testing purposes, also create a default user and password for each of these groups. In addition, if you want to follow the Decision Center permissions tutorial in your own installation, you must create the two `Validator` and `Eligibility` custom groups.

The following table summarizes the mandatory and custom groups and their associated role, default user, and password.

| Group            | Use                                                                      | Default User - Password |
|------------------|--------------------------------------------------------------------------|-------------------------|
| rtsAdministrator | Mandatory, gives the user administrator access.                          | rtsAdmin - rtsAdmin+0   |
| rtsConfigManager | Mandatory, gives the user configuration manager access.                  | rtsConfig - rtsConfig+0 |
| rtsUser          | Mandatory, gives a user standard access.                                 | rtsUser1 - rtsUser1+0   |
| rtsInstaller     | Mandatory, gives the user access to the Installation Settings wizard.    | rtsAdmin - rtsAdmin+0   |
| Validator        | Optional custom group, used in the Decision Center permissions tutorial. | Val - ValVal+0          |
| Eligibility      | Optional custom group, used in the Decision Center permissions tutorial. | Eli - EliEli+0          |

**Note:** For more information on the Decision Center groups, see [Groups](#).

You declare groups and users by using the JBoss **add-user** command.

The following procedure is a suggested configuration with a default user for each of the basic groups.

### Procedure

In the `<JBoss_HOME>/bin` directory, open a command prompt and run the following **add-user** commands:

```
./add-user[.sh|.bat] -a -u rtsAdmin -p rtsAdmin+0 -ro "rtsAdministrator,rtsInstaller,rtsUser"
./add-user[.sh|.bat] -a -u rtsConfig -p rtsConfig+0 -ro "rtsConfigManager,rtsUser"
./add-user[.sh|.bat] -a -u rtsUser1 -p rtsUser1+0 -ro "rtsUser"
./add-user[.sh|.bat] -a -u Val -p ValVal+0 -ro "Validator,rtsUser"
./add-user[.sh|.bat] -a -u Eli -p EliEli+0 -ro "Eligibility,rtsUser"
```

### What to do next

After you deploy the archive in the next step, you must upload all groups to the database from the Installation Settings wizard of the Decision Center console. For more information, see “Step 3: Set up groups” on page 72.

### Declaring custom groups

If you create custom groups, you must declare them before you deploy the EAR file.

### About this task

The Decision Center EAR file references the basic groups: `rtsUser`, `rtsConfigManager`, `rtsAdministrator`, and `rtsInstaller`.

You must add any custom groups that you declared in “Defining groups and permissions” on page 66. Also, add the `Validator` and `Eligibility` groups that you created for the Decision Center tutorials.

**Tip:**

- The EAR file is compressed. You must open it to extract the files that must be changed, and then replace the files in the EAR file. You can use Ant commands to repackage the EAR file, as explained in “Repackaging the Decision Center archive” on page 83.
- Back up the EAR file before you modify it.
- To use the Decision Center permissions mechanism, you must upload groups to the database. For more information, see “Completing the configuration from the Decision Center Enterprise console” on page 69.

**Procedure**

1. Add each custom group as a role in the appropriate application.xml file.  
<ODM\_InstallDir>/teamserver/applicationservers/JBoss6EAP/jrules-teamserver-JBOSS61EAP.ear/META-INF/ For example:  
...  
<security-role>  
  <role-name>my\_custom\_group</role-name>  
</security-role>  
...  
2. If you have already deployed the archive, upload the custom groups to the database from the Installation Settings wizard of the Decision Center console. For more information, see “Step 3: Set up groups” on page 72.

**What to do next**

Otherwise, proceed to “Step 4: Deploying the Decision Center EAR on JBoss” and upload the groups when you complete the configuration.

## Step 4: Deploying the Decision Center EAR on JBoss

To deploy the Decision Center EAR on JBoss, you must copy the EAR archive to the deployment directory of your application server.

**Procedure**

Copy the file <ODM\_InstallDir>/teamserver/applicationservers/JBoss6EAP/jrules-teamserver-JBOSS61EAP.ear to the deployments folder:  
<JBOSS\_HOME>/standalone/deployments

**Important:**

Deploying the Decision Center EAR sets the persistence locale. After you save a rule to the database, you cannot change the persistence locale. If you want to install Decision Center in a language other than English, take note of the instructions provided in “Step 6: Completing your Decision Center configuration” on page 69.

## Step 5: Verifying the deployment of the Decision Center consoles

After you have finished configuring Decision Center for your application server, verify that you have deployed the archive successfully.



## About this task

You start your application server and then use your web browser to open the Decision Center Enterprise console.

### Procedure

1. Make sure that your application server is running.
2. Start a new browser instance and enter the default URL to access Decision Center in a web browser.
  - The default URL to access Decision Center is `http://localhost:<PORT_NUMBER>/teamserver`. Set `<PORT_NUMBER>` to the port number of your web application. The default value of the port number is 8080. If your browser is not running on the same host as the application server, replace `localhost` with the address of the host.
  - Starting from WebSphere Application Server V8.0, each console has a different default URL.
    - For the Enterprise Console: `http://localhost:<PORT_NUMBER>/teamserver`
    - For the Business Console: `http://localhost:<PORT_NUMBER>/decisioncenter`
  - If your browser is not running on the same host as the application server, replace `localhost` with the address of the host.
  - If your web application is mapped to a host on a port that is different from the port number shown, change the port number to your host port number.The Decision Center log in page opens in your browser.
3. Sign in with `rtsAdministrator` rights to start testing.  
For example, `rtsAdmin` and `rtsAdmin+0`.

## Step 6: Completing your Decision Center configuration

After you have created a data source, defined security settings, and deployed the Decision Center EAR file, you complete the configuration either from the Decision Center console or by running Ant tasks. If your database is DB2 for z/OS, generate the schema on z/OS for reasons of performance. You can, however, use the wizard to import extension files.

### Completing the configuration from the Decision Center Enterprise console

After you have deployed the Decision Center archive to your application server, you can work from the Decision Center Enterprise console to complete or modify the configuration.

#### Installation Settings wizard overview:

You use the Installation Settings wizard in the Decision Center console to create or modify the database schema, set up message files or groups, or change the persistence locale or configuration parameters.

The Installation Settings wizard opens automatically when you start the Decision Center console to complete an installation.

You can also open the Installation Settings wizard by clicking **Configure > Installation Settings Wizard** in the Decision Center console after you have completed your initial installation. If you open Decision Center after you have

followed the steps to install the module, only the **Install** tab is available. For more information, see *Opening the Decision Center consoles*.

**Note:** To access the Installation Settings wizard, you must have both administrator privileges and the `rtsInstaller` role when you sign in.

You use the Installation Settings wizard for the following actions.

*Table 5. Actions in the Installation Settings wizard*

| Action                          | Description                                                                                                                                                                                              |
|---------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Configure the database          | Mandatory when you complete the configuration with a database on a distributed platform. For more information, see Step 1: Configure the database.                                                       |
| Set up message files            | Mandatory during the installation only if you have some custom rule model extension files. For more information, see Step 2: Set up message files.                                                       |
| Set up groups                   | You must set up the same groups that are declared in the application server if you want to use the Decision Center security and permissions mechanisms. For more information, see Step 3: Set up groups. |
| Change the persistence locale   | Mandatory if the persistence locale is different from the locale <code>en_US</code> . For more information, see Step 4: Set the persistence locale.                                                      |
| Change configuration parameters | Optional. You change some configuration parameters when you customize Decision Center. For more information, see Step 5: Set configuration parameters.                                                   |

After you have completed the installation, Decision Center is ready to use but does not contain rule projects. If you open Decision Center and no rule projects are available, a message in the **Configure** tab informs you that no project has been found and that you should either publish a rule project by using Rule Designer or contact the administrator.

If you see this message, you must publish a rule project from Rule Designer. For more information, see *Publishing a project to Decision Center*.

More information about using the Installation Settings wizard is available from the Decision Center console online help. To access the online help, click **Help** in the top banner after you have signed in to Decision Center.

**Note:** If you have rule projects that were created with a previous product version, refer to the Migrating topics on how to upgrade the Decision Center database schema.

### Step 1: Configure the database:

You use the Installation Settings wizard to configure the database.

#### About this task

You store the extensions to the Decision Center rule model in two XML files:

- Model description: This file usually has the file name extension `.brmx`.
- Initialization of enumerations and hierarchies: This file usually has file name extension `.brdx`.

For more information about defining common model extensions, see the customization topics.

### Procedure

1. When the Installation Settings wizard opens in Decision Center, click **Next**.
2. Select one of the extension files.
  - **Default extensions** (already selected)
  - **Custom extensions (brmx/brdx)**, or
  - **Custom extensions (Zip)**
3. Click **Generate SQL** to generate the script that creates the database tables, which are based on the contents of your rule model files.
4. After the script is generated, select the **Execute the SQL script** check box, and then click **Next**.

### Step 2: Set up message files:

Message files contain the display text that is associated with the extensions to the rule model contained in the .brmx and .brdx files.

### About this task

You can find the default message file in `<ODM_InstallDir>/teamserver/bin/defaulttextextensionmessages_<LOCALE>.properties`.

If you use the default rule model when you create your database, the default message file is automatically sent to the database. To upload your own message files, use the Installation Settings wizard as explained below.

You must have a message file for each locale that you use. Message files are identified by their locale. The contents of the message files must respect the ISO-LATIN-1 standard.

### Procedure

To declare a message file in the Installation Settings wizard:

1. Click **New**.
2. Enter a locale.
3. Browse to the location of the message file for this locale.
4. Click **Apply**.

### Results

If Decision Center supports this locale, the Installation Settings wizard assigns a locale code so that you can identify it.

### Example

For example:

```
status=Status
effectiveDate=Effective Date
expirationDate=Expiration Date
new=New
defined=Defined
```

### Step 3: Set up groups:

In addition to creating groups in your application server when you set up security access, you must use the Setup Groups page in the Installation Settings wizard to upload groups to the database.

#### Before you begin

Before you set up groups in the Enterprise console, make sure to add all the groups that you want to see in the available list when enforcing project security or setting permissions in Decision Center. For more information, refer to the topics on Groups and Permissions in the Decision Center help.

#### About this task

You set up groups only if you want to use the Decision Center project access and permission mechanisms.

**Tip:** In Decision Center, the groups are the roles in the application server, **not** the groups defined in the user registry. Decision Center uses the group information to verify whether a user belongs to a role in the application server.

You do not have to upload the `rtsAdministrator` or `rtsInstaller` group. The administrator group has access to everything, and an installer user must belong to another group.

You use the Setup Groups page in the Installation Settings wizard to upload the default groups for `rtsUser` and `rtsConfigManager`, and any custom groups, such as `Validator` and `Eligibility` if you want to follow the permissions tutorial.

#### Procedure

To set up groups:

1. Click **New**.
2. Type the group name.
3. Click **Apply**.
4. Repeat steps 1 on page 45 to 3 on page 45 for each group.
5. When you have added all the groups, proceed in one of the following ways:
  - Click **Next** if you want to set a different persistence locale, or configuration parameters.
  - Click **Finish** if you do not want to change these settings.

### Step 4: Set the persistence locale:

The persistence locale determines the language in which you store rules in the Decision Center database.

#### About this task

You set the locale when you deploy the Decision Center EAR file to your application server. As a consequence, you store the rules in the database in the locale of the Decision Center application.

Changing the persistence locale does not change the language in which Decision Center displays rules. Changing it in Decision Center is necessary only to match the locale of Rule Designer when you synchronize your rule projects, and to access the tutorials in your locale.

**Important:** You must not change the persistence locale after you have saved a rule to the database.

### Procedure

To set the persistence locale:

1. Enter a locale in the **Locale** field.
2. Click **Apply**.
3. Proceed as follows:
  - Click **Next** if you want to set the configuration parameters.
  - Click **Finish** if you do not want to change these settings.

### Step 5: Set configuration parameters:

Many tasks that are related to customizing Decision Center require that you add or remove configuration parameters.

### About this task

Decision Center uses the following configuration parameters to generate complete URLs in permalinks:

- **teamserver.server.port:** the port number
- **teamserver.server.isSecure:** true if the connection is secure
- **teamserver.server.hostname:** the name of the host.

To create, modify, or delete configuration parameters, you use the Set configuration parameters page in the Installation Manager wizard. You generate these parameters when you sign in to the Decision Center console for the first time after you have configured the database. You can use the Installation Settings wizard to change these parameters at any time.

The following table gives a description of the main configuration parameters available in `teamserver.war/WEB-INF/lib/teamserver-model-XXX.jar/ilog/rules/teamserver/preferences.properties`.

### Note:

The parameters in the table include the **teamserver** prefix, which is not in the `preferences.properties` file. You must include the prefix when you set configuration parameters in the Installation Settings wizard.

| Parameter                                                | Used to                                                                                                                                                                                                                                                                                             |
|----------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>teamserver.&lt;extractorValidator&gt;.class</code> | Specify a class of ruleset extractor validators to use for the <code>extractorValidator</code> name. The class must implement the <code>IlrExtractorValidator</code> interface. After you define this class, specify this name as the extractor validator to use when defining a ruleset extractor. |

| Parameter                                        | Used to                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|--------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>teamsserver.build.path</b>                    | Define the location of the IRL cache in the file system. Compute the path as follows: <ul style="list-style-type: none"> <li>• Use this property with the name of the user who started the server as the root for the cache (<i>&lt;build.path&gt;_&lt;username&gt;</i>).</li> <li>• If this property is not defined, use the system property <b>java.io.tmpdir</b> and add <i>rtscache</i>. For example, <i>&lt;temp dir&gt;/rtscache_&lt;username&gt;</i>.</li> <li>• If the system property is not defined, use the server directory and add <i>rtscache</i>. For example, <i>&lt;server dir&gt;/rtscache_&lt;username&gt;</i>.</li> </ul> |
| <b>teamsserver.br1.verbalizers</b>               | Specify the list of locales for which a BAL verbalizer is defined.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| <b>teamsserver.br1.verbalizer.&lt;locale&gt;</b> | Specify the verbalizer class for the locale. The class must implement the <code>ilog.rules.vocabulary.verbalization.IlrVerbalizer</code> interface.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |

### Procedure

1. Create a parameter or change an existing one.
  - To create a parameter, click **New**.
  - To change a parameter, select the check box next to the parameter and then click **Modify** to change the parameter or click **Delete** to remove it.
2. Click **Apply** to implement your changes.
3. Proceed as follows:
  - Click **Previous** if you want to make changes to previous settings.
  - Click **Finish**. The Installation log opens with a summary of the operations that you performed in the Installation Settings wizard.
4. Click **OK** to finish.

### What to do next

You now have to sign in to the Decision Center console. Continue with the section Publishing a project.

## Completing the configuration by using Ant tasks

Ant tasks provide an alternative method for completing or modifying the configuration. These tasks perform the same configuration steps as the Installation Settings wizard in the Decision Center console.

### Setting up the Ant tasks environment:

To run Decision Center Ant tasks, you must first set up the required environment variables.

To run Ant tasks, you must have version 1.7.1 (or later) of Ant set up on your system. If Ant is not installed or your version is older than version 1.7.1, you must set up your environment to use the correct version of Ant. You can download Ant from the Apache web site, or you can use the Ant 1.7.1 distribution packaged at `<ODM_InstallDir>/shared/tools/ant`, where `<ODM_InstallDir>` is your Operational Decision Manager installation directory.

### To test your current version of Ant:

Type the following command in a Windows command prompt or UNIX shell: `ant -version`

### To set up your environment to use Ant:

1. Set the `ANT_HOME` environment variable to `<ODM_InstallDir>/shared/tools/ant`.
2. Set the `JAVA_HOME` environment variable to the path to your JDK installation (1.6).
3. Add the directory `<ODM_InstallDir>/shared/tools/ant/bin` to your `PATH` environment variable.

The Decision Center Ant tasks are defined in `<ODM_InstallDir>/teamserver/bin/build.xml` and executed by commands of the form:

```
ant <taskName> <parameters list>
```

**Note:** To execute these Ant tasks, you must use the same Java virtual machine version and vendor as the one that is used by the application server.

Ant task parameters start with `-D`. Use them to set values such as the following ones:

- `-Dserver.url=<server url>`: Specifies the URL of the target application server.
- `-DdatasourceName=<data source name>`: Specifies the JNDI name of the data source to use for the task. The default value is `jdbc/ilogDataSource`.

```
ant execute-schema -Dserver.url=<protocol://host:port>/teamserver/
-DdatasourceName=jdbc/ilogDataSource -Dfile=my_sql_file.sql
```

The `<protocol://host:port>` URL is defined in the file `<ODM_InstallDir>/teamserver/bin/teamserver-anttasks.properties`. If your browser is not running on the same host as the application server, replace `localhost` with the address of the machine. If your web application is mapped to a host on a port that is different from the port number shown, change the port number to your host port number.

The file `<ODM_InstallDir>/teamserver/bin/teamserver-anttasks.properties` defines the value of some common parameters and others that depend on the application server used. You do not have to include these parameters in your Ant task command if they are properly defined in this file. The content of the `teamserver-anttasks.properties` file is as follows:

```
Default properties

rtsAdmin.login=rtsAdmin
rtsAdmin.password=rtsAdmin

protocol=http
server.host=localhost
server.port=8080
server.url=${protocol}://${server.host}:${server.port}/teamserver

datasourceName=jdbc/ilogDataSource

outputFile=output.sql

languagePackPath = .
languagePackOutputPath = ./generated
```

```
persistenceLocale =
selector =
branch =
override = false
```

Also, take note of any special instructions in this file concerning your application server.

The **appserver.name** property configures the class path for the Ant tasks. If you have to add specific drivers to your class path, you can add them to `<ODM_InstallDir>/teamserver/lib/classpath-teamserver.xml`.

Communication between the Ant tasks and Decision Center supports the HTTP or HTTPS communication protocols. For more information, see Ant task communication protocol.

### Creating the database schema:

You can create the database schema in a single operation by using the **set-extensions** Ant task, or choose to create it step by step.

*Creating schemas with the set-extensions Ant task:*

For convenience, you can create the database schema by using the **set-extensions** Ant task.

### About this task

Extensions to the Decision Center rule model are stored in two XML files.

- Model description: This file usually has the file name extension `.brmx`.
- Initialization of enumerations and hierarchies: This file usually has file name extension `.brdx`.

You can use Ant tasks to load the rule model from the two XML files and build the SQL script that is necessary to get the proper database schema.

**Note:** To run these Ant tasks, you must use the same Java virtual machine version and vendor as the one used by the application server.

Alternatively, you can create the database schema step by step, which is useful if you want to look at the generated SQL schema. See “Creating the schema using a step-by-step sequence” on page 50.

### Procedure

Run the **set-extensions** Ant task.

This Ant target runs **gen-create-schema** + **execute-schema** + **upload-extensions** + **upload-roles**. Set the following parameters:

**-Dserver.url**=<server url>

**-DdatasourceName**=<data source name>

**-DextensionModel**=<model file>

The model description file, with the `.brmx` extension

**-DextensionData**=<data file>

The model data description, with `.brdx` extension



**[-DdbName=<database schema name>]**

You can use this optional parameter to specify the database schema name. Otherwise, Decision Center uses the database user name as the schema name. However, some databases allow some users to access several schemas, and the default schema does not always reflect the user name.

**[-Droles=<role list>]**

You can use this optional parameter to upload the list of roles to Decision Center. This list is specified as "role1 role2". For example:

```
ant upload-roles -Droles="rtsUser rtsConfigManager Eligibility Validator"
```

*Creating the schema using a step-by-step sequence:*

If you want to look at the generated SQL schema, you can create it step-by-step.

*Creating the database schema script:*

You can create the database schema script by using the **gen-create-schema** Ant task.

### Procedure

To create the SQL script that is necessary to create or update the database schema, run the **gen-create-schema** Ant task with these parameters:

**-Dserver.url**=<server url>

**-DdataSourceName**=<data source name>

**-DextensionModel**=<model file>

The model description (.brmx extension).

**-DextensionData**=<data file>

The model data description (.brdx extension).

**[-DdbName=<database schema name>]**

You can use this optional parameter to specify the database schema name in which the Decision Center tables are stored. If you do not specify the parameter, Decision Center uses the database user name as the schema name. However, some databases allow some users to access several schemas, and the default schema is not always named the same as the user.

**[-DoutputFile=<SQL file>]**

The name of the file that stores the generated SQL script. If this parameter is not specified, the task creates a file named output.sql in the directory that is defined as basedir in the build.xml file.

```
ant gen-create-schema -DextensionModel=my_model_file.brmx -DextensionData=my_data_file.brdx -Dout
```

### Results

1. The task connects to the specified data source from the application server.
2. The task checks whether this data source points to an existing Decision Center database.
  - If a database does not exist, the task builds the SQL script to create a fresh database schema to store the model.
  - If a database does exist, the task builds the SQL script that is necessary to update the existing database schema.

*Executing the database schema script:*

You execute the database schema script.

### Procedure

To execute the SQL script that you created, run the **execute-schema** Ant task with these parameters:

**-Dserver.url**=<server url>

**-DdatasourceName**=<data source name>

**[-Dfile**=<SQL file>]

The name of the file to execute, which corresponds to the script that you created. If you do not specify this parameter, the task attempts to execute a file named `output.sql` in the directory that is defined as `basedir` in the `build.xml` file.

```
ant execute-schema -Dfile=my_sql_file.sql
```

*Uploading the database schema extension:*

You upload the database schema extension.

### Procedure

To store the rule model description in the database schema, run the **upload-extensions** Ant task with these parameters:

**-Dserver.url**=<server url>

**-DdatasourceName**=<data source name>

**-DextensionModel**=<model file>

The model description (`.brmx` extension).

**-DextensionData**=<data file>

The model data description (`.brdx` extension).

```
ant upload-extensions -DextensionModel=my_model_file.brmx -DextensionData=my_data_file.brdx
```

### Results

The description is stored in the database so that Decision Center applications can load it when they start. It is also used by **gen-create-schema** to get the current model description to run a `diff` with the new schema.

In a cluster, you must restart the servers and close all current sessions.

*Uploading a list of roles or groups to the database:*

In addition to creating groups in your application server when you set up security access, you must upload groups to the database.

### Before you begin

1. Add all the groups that you want to see in the available list when you enforce project security or setting permissions in Decision Center.
2. Create the default groups for `rtsUser` and `rtsConfigManager`, and upload your custom groups.

## About this task

You must upload roles or groups only if you want to use the Decision Center project access and permissions mechanisms. For more information, see the topics on Groups and Permissions in the Decision Center online help.

You do not have to upload the `rtsAdministrator` group or the `rtsInstaller` group. The Administrator group has access to everything and an Installer user must belong to another group.

**Note:** To run these Ant tasks, you must use the same Java virtual machine version and vendor as the one used by the application server.

## Procedure

To store in the database the list of roles or groups to be used by the application, run the **upload-roles** Ant task with the following parameters:

**-Dserver.url**=<server url>

**-DdataSourceName**=<data source name>

**-Droles**=<role list>

<role list> is the list of roles or groups to upload to Decision Center, specified as "group1 group2".

```
ant upload-roles -Droles="rtsUser rtsConfigManager Eligibility Validator"
```

*Removing a database schema:*

You can create an SQL script to remove (drop) a database schema by using the **gen-drop-schema** Ant task.

## About this task

To remove a database schema, you proceed in two steps:

1. Create the SQL script that is necessary to remove the database schema.
2. Execute the SQL script that you created.

## Procedure

1. To create the SQL script to delete a database schema, run the **gen-drop-schema** Ant task with the following parameters:

**-Dserver.url**=<server url>

**-DdataSourceName**=<data source name>

**-DextensionModel**=<model file>

The description of the database schema to remove.

**[-DdbSchemaName=<database schema name>]**

You can use an optional parameter to specify the database schema name. If you do not specify this parameter, Decision Center uses the database user name as the schema name. However, in some databases, users can access several schemas and the default schema is not always named as the user.

**[-DoutputFile=<SQL file>]**

The name of the file that stores the generated SQL script. If you do not specify this parameter, the task creates a file named `output.sql` in the directory that is defined as `basedir` in the `build.xml` file.

```
ant gen-drop-schema -DextensionModel=my_model_file.brmx -DoutputFile=my_sql_file.sql
```

2. To execute the SQL script that you created, run the **execute-schema** Ant task with these parameters:

```
-Dserver.url=<server url>
```

```
-DdataSourceName=<data source name>
```

```
[-Dfile=<SQL file>]
```

The name of the file to execute, which corresponds to the script that you created. If you do not specify this parameter, the task attempts to execute a file named `output.sql` in the directory that is defined as `basedir` in the `build.xml` file.

```
ant execute-schema -Dfile=my_sql_file.sql
```

The task connects to the specified data source from the application server. It reads the model description that is passed in the parameters, and generates the SQL script to remove the existing schema. Because many database tables are linked through foreign keys, these tables must be removed in a specific order and the script generation handles these constraints.

### Example

Here is the complete code sample:

```
ant gen-drop-schema -DextensionModel=my_model_file.brmx -DoutputFile=my_sql_file.sql
ant execute-schema -Dfile=my_sql_file.sql
```

### Defining and uploading message files:

You can define and upload message files to Decision Center by using the **upload-messages** Ant task.

Message files contain the display text that is associated with the extensions to the rule model that is contained in the `.brmx` and `.brdx` files. For example:

```
status=Status
effectiveDate=Effective Date
expirationDate=Expiration Date
new=New
defined=Defined
```

The default messages file is provided in: `<ODM_InstallDir>/teamserver/bin/defaulttextensionmessages_<LOCALE>.properties`

**Note:** The contents of the messages files must conform to the ISO-LATIN-1 standard.

You must have a messages file for each locale that you use. Upload the messages file to Decision Center by running the **upload-messages** Ant task with these parameters:

- **-Dserver.url=<server url>**
- **-DdataSourceName=<data source name>**
- **-Dlocale=<locale>**
- **-DmessageFile=<message file>**

```
ant upload-messages -Dlocale=en_US -DmessageFile=mymessages.properties
```

## Setting the persistence locale:

The persistence locale is used to determine the language in which rules are stored in the Decision Center database.

### About this task

The persistence locale is set when you deploy the Decision Center archive to your application server, which means that the rules in the database are stored in the locale of the Decision Center application.

Changing the persistence locale does not change the language in which rules display in Decision Center. Changing the persistence locale in Decision Center is necessary only to match the locale of Rule Designer when you synchronize your rule projects, and to access the tutorials in your locale.

**Important:** You must not change the persistence locale after you have saved a rule to the database.

### Procedure

To set the persistence locale by running an Ant task:

1. Open the `<ODM_InstallDir>/teamserver/bin/teamserver-anttasks.properties` file.

This file defines the value of some common parameters.

2. Add your locale to the **persistenceLocale** property and save the `teamserver-anttasks.properties` file.

For example: `persistenceLocale = fr_FR`

3. Run the Ant task in this form: `ant taskName parameters_list`

Alternatively, you can add the parameter to the command line. For example:

```
ant taskName -DpersistenceLocale=fr_FR
```

### Adding or removing configuration parameters:

For many tasks that are related to customizing Decision Center, you must add or remove configuration parameters.

The following configuration parameters, used to generate complete URLs in permalinks, are generated the first time you sign in to Decision Center after the database is configured. You can use the Installation Settings wizard to set these parameters beforehand or change them afterward:

- **teamserver.server.port:** The port number
- **teamserver.server.isSecure:** true if the connection is secure.
- **teamserver.server.hostname:** The name of the host

The following table gives a description of the main configuration parameters available in `teamserver.war/WEB-INF/lib/teamserver-model-XXX.jar/ilog/rules/teamserver/preferences.properties`.

| Parameter                                                 | Use                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|-----------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>teamservice.&lt;extractorValidator&gt;.class</code> | Specify a ruleset extractor validator class to use for the <code>extractorValidator</code> name. The class must implement the <code>ILrExtractorValidator</code> interface. After this class is defined, specify this name as the extractor validator to use when defining a ruleset extractor.                                                                                                                                                                                                                                                                                                                                                                                                           |
| <code>teamservice.build.path</code>                       | Define where the cache of the IRL is located on the file system. The path is computed as follows: <ol style="list-style-type: none"> <li>1. First, use this property with the name of the user who started the server as the root for the cache:<br/><code>&lt;build.path&gt;_&lt;username&gt;</code>.</li> <li>2. If the path is not defined, use the system property <code>java.io.tmpdir</code> and add <code>rtscache</code>. For example,<br/><code>&lt;temp_dir&gt;/rtscache_&lt;username&gt;</code>.</li> <li>3. If the system property is not defined, use the server directory and add <code>rtscache</code>. For example, <code>&lt;server_dir&gt;/rtscache_&lt;username&gt;</code>.</li> </ol> |
| <code>teamservice.br1.verbalizers</code>                  | Specify the list of locales for which a BAL verbalizer is defined.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <code>teamservice.br1.verbalizer.&lt;locale&gt;</code>    | Specify the verbalizer class for the specified locale. The class must implement the <code>ILrVerbalizer</code> interface.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |

You can use the following Ant tasks to add or remove configuration parameters:

#### set-config-param

Sets a configuration parameter for a specified user. If the user is not specified, it sets a *global parameter*.

Parameters:

- `-Dserver.url=<server url>`
- `-DdatasourceName=<data source name>`
- `[-Duser=<username>]`
- `-Dkey=<parameter key>`
- `-Dvalue=<parameter value>`

For example:

```
ant set-config-param -Dkey=locale -Dvalue=en_US
```

#### remove-config-param

Drops the configuration parameter for a specified user. If the user is not specified, it drops the global configuration parameter.

Parameters:

- `-Dserver.url=<server url>`
- `-DdatasourceName=<data source name>`
- `[-Duser=<username>]`
- `-Dkey=<parameter key>`

### print-config-param

Prints the global parameters or specified user parameters if the *username* value is specified. If no key is specified, all keys are printed.

Parameters:

- **-Dserver.url=<server url>**
- **-DdataSourceName=<data source name>**
- **[-Duser=<username>]**
- **-Dkey=<parameter key>**

### Repackaging the Decision Center archive:

You can repack the Decision Center archive by using an Ant task.

When you add new .jar files to the Decision Center archive, you must repack the archive by running the **repackage-ear** or **repackage-war** Ant task. This task does not use the **server.url** and **dataSourceName** parameters.

If you have customized Decision Center, you must package the custom .jar files before you use the Ant task to repack the Decision Center.

The **repackage-ear** or **repackage-war** Ant task takes the following parameters:

**-DtargetEar=<target ear>**

**-DtargetWar=<target war>** for the Tomcat application server

**-DsourceEar=<source ear>**

**DsourceWar=<source war>** for the Tomcat application server

**-DdescriptorsDir=<descriptors directory>**

A directory that is copied into the META-INF directory of the target EAR (not mandatory).

**-DadditionalJars=<"myjar1.jar,myjar2.jar, myjarn.jar">**

Additional .jar files to store in the lib directory of the target archive (not mandatory).

**-DtmpDir=<directory>**

A directory that you can specify to store temporary files (not mandatory).

**-DwebResourcesDir=<web resources directory>**

A directory that is copied into the WAR library (not mandatory).

**-Dconsole=both|enterprise|business**

Specifies whether to repack the Business or the Enterprise WAR files. The default is both.

## Configuring the search function of the Decision Center Business console

The search function in the Decision Center Business console is based on the Solr search engine. You must configure the engine to provide this functionality.

### Using a remote Solr search engine

You can set up the search function in Decision Center Business console to work with a remote instance of the Apache Solr search engine.

## About this task

The search function in Decision Center Business console uses an embedded instance of the Apache Solr search engine. Alternatively, you can have the search function work with a remote instance of the search engine, which you can run on another computer, or the same computer but in a dedicated web application.

## Procedure

To configure the search function to run with a remote instance of the Solr search engine:

1. Install the Apache Solr search engine on another computer, or as part of a dedicated web application on your computer.  
For information on installing the Solr server, visit the Apache Solr website.
2. Locate the `decisioncenter-solr-home.zip` file in the Decision Center `teamserver` folder on your computer.  
The Decision Center installation program placed the folder on your computer.
3. Decompress the file in a directory in the remote instance of the Solr server.
4. Configure the home directory of the remote Solr server to use the location of the decompressed configuration files.  
For more information, see the documentation on the Apache Solr website.
5. Configure the Decision Center `preferences.properties` file to point to the URL of the Solr server.

## Setting parameters for the Solr search engine

You can configure the search function in the Decision Center Business console to run with the Apache Solr search engine.

Decision Center Business console provides a function for searching rule projects. When you install this feature, it uses an embedded instance of the Apache Solr search engine. Alternatively, you can have the search function run with a remote instance of the search engine, which you can install on another computer, or on the same computer but in a dedicated web application. For more information, see “Using a remote Solr search engine” on page 56.

You can set three parameters for the search engine:

Table 6. Search parameters

| Parameter                               | Description                                                                                                                                                                                                                                                                                       |
|-----------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>search.SearchProvider</code>      | This parameter takes one of the following values: <ul style="list-style-type: none"><li>• <code>SolrEmbedded</code>: Use this value to select the embedded Solr search engine.</li><li>• <code>SolrRemote</code>: Use this value to select a remote instance of the Solr search engine.</li></ul> |
| <code>search.SolrEmbeddedDataDir</code> | Use this optional parameter to direct the index of the embedded version ( <code>SolrEmbedded</code> ) to a specific directory on the Decision Center.                                                                                                                                             |
| <code>search.SolrRemoteUrl</code>       | Use this parameter with <code>SolrRemote</code> to provide the URL of the remote Solr search engine.                                                                                                                                                                                              |



You set the configuration parameters for the search engine within the `preferences.properties` file for Decision Center. For more information on the `preferences.properties` file, see Configuration parameter maintenance.

The following table provides examples for setting the parameters in `preferences.properties` to work with the Solr search engine:

Table 7. Configuration table for search properties

| Solr server                                                                                                                                                                                          | Parameter settings                                                                                                                                                   |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>External server</p> <p>The URL of the external server depends on the installation. For this example, the remote address of the search engine is <code>http://mysearchserver:8983/solr</code>.</p> | <p>Set the <code>preferences.properties</code> file as follows:</p> <pre>search.SearchProvider=SolrRemote search.SolrRemoteUrl=http://mysearchserver:8983/solr</pre> |
| <p>Embedded server</p> <p>In this example, you store the index in the <code>c:/temp/DC-SearchIndex</code> directory on your computer.</p>                                                            | <p>Set the <code>preferences.properties</code> file as follows:</p> <pre>search.SearchProvider=SolrEmbedded search.SolrEmbeddedDataDir=c:/temp/DC-SearchIndex</pre>  |

If you do not specify a directory for `search.SolrEmbeddedDataDir`, Decision Center stores the search index in a temporary directory. When the server stops running, it also stops using the temporary directory. When the server restarts, it creates a new temporary directory, and completely re-indexes the repository.

If you specify a directory for `search.SolrEmbeddedDataDir`, the directory and its content persist across server restarts, and the server does not re-index the repository with each restart.

## Additional steps to configure Decision Validation Services

There are additional steps that you must perform to configure Decision Validation Services.

### Before you start

Before deploying and configuring Decision Validation Services, you must complete some steps.

1. Install Rule Execution Server.
2. Install Decision Center.
3. Configure both of these applications on the same application server.

After these installations are complete, see “What steps to follow” on page 58 to understand the additional tasks that you might have to complete.

### Note:

Only users belonging to the `resAdministrators` or `resMonitors` groups are able to see the Decision Warehouse tab in the Rule Execution Server console after Decision Validation Services is configured.

### What steps to follow

The configuration steps you follow to configure Decision Validation Services are summarized in the following table:

| Step                                                                                                                                                                                                | Required                                                                                                                                                                                                                                         |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| "Step 1: Creating Decision Warehouse database resources" on page 59                                                                                                                                 | Optional. If you have used the Rule Execution Server console to create database resources, you do not have to perform this task. See Creating a database schema by using the Rule Execution Server console for more information.                 |
| "Step 2: Deploying the Decision Validation Services archive for the first time" on page 61<br><br>This archive is the default SSP archive packaged with the Decision Validation Services installer. | Optional. This steps allows you to check the availability of the feature.<br><b>Note:</b><br><br>You can also use the default SSP archive if you have an XML XOM. In this case, you do not have to repackage the archive to include the XML XOM. |
| "Step 3: Checking the availability of Decision Validation Services" on page 61                                                                                                                      | Optional.                                                                                                                                                                                                                                        |
| "Step 4: Packaging Decision Validation Services by using Ant" on page 62                                                                                                                            | Optional. You can use this task as an alternative method for configuring or modifying the Decision Validation Services installation.                                                                                                             |

## Step 1: Creating Decision Warehouse database resources

You can use SQL scripts to create a dedicated schema in the database.

If you set the Rule Execution Server persistence to `datasource` or `jdbc` and you intend to use Decision Warehouse to store your test execution results, you must create a dedicated schema in the database that contains these tables and views. To do so, you can use the provided SQL scripts, located in `<ODM_InstallDir>/executionserver/databases`. A readme file in this directory provides additional information about the scripts.

The script that creates the Decision Warehouse database schema is named `trace_<database_name>.sql`.

### Note:

The Installation Settings wizard in the Rule Execution Server console creates all the necessary tables for Rule Execution Server and for Decision Warehouse. If you are configuring Decision Validation Services and you have already run the Installation Settings wizard to create the tables, you do not have to create database resources manually. However, if you did not use the wizard to create database resources, you must run the script to create the Decision Warehouse database schema.

When you use DB2 (except on z/OS), the scripts that create the Rule Execution Server database tables are written for databases that use automatic storage.

- BP32K is the buffer pool that is expected in SYSCAT.BUFFERPOOLS. If BP32K is not there, you can use the existing buffer pool or create a new buffer pool named BP32K. Use the following command to query SYSCAT.BUFFERPOOLS for the existing buffer pool:

```
Select * from SYSCAT.BUFFERPOOLS
```

Otherwise, use the following command to create a buffer pool named BP32K:

```
CREATE BUFFERPOOL BP32K SIZE 2000 PAGESIZE 32K
```

- You must update the trace\_db2.sql script and select the custom option in the Installation Settings wizard to run it. Modify the following line in the script to specify storage for the table space:

```
CREATE TABLESPACE RESDWTS PAGESIZE 32K BUFFERPOOL BP32K;
```

Here is an example of the table space specification in the script:

```
CREATE TABLESPACE RESDWTS PAGESIZE 32K MANAGED BY Database
USING [FILE 'C:\DB2\Container.file' 640] BUFFERPOOL BP32K;
```

- You might have to further modify the script based on your database settings.

**Note:**

If you completed this task as part of the Rule Execution Server configuration, you do not have to do this a second time. It is included here solely for completeness.

If you use Command Editor to run the scripts, you must log in with the credentials that you use for the data source for Rule Execution Server.

Use any tool that can handle SQL to import and run the SQL scripts. The tools provided for each database include:

| Database    | Database tool                  |
|-------------|--------------------------------|
| IBM DB2     | DB2 command line processor     |
| Derby       | ij command line processor      |
| MySQL       | mysql command line processor   |
| Oracle      | sqlplus command line processor |
| Postgre SQL | Postgre SQL command line tool  |
| SQL Server  | Query Tool                     |
| Sybase      | isql command line processor    |

To access the database, the database user must have the following credentials:

- A user ID and a password
- Complete privileges on the tables and view of the schema (create, insert, delete)
- create index privileges
- On Oracle, create trigger and create sequence privileges. If you use an Oracle database, run all the scripts in the SQL Plus client.

Install a database client for the database that you use. Refer to the documentation of the database for more information.

The default CLOB size might not be sufficient for the FULL\_EXECUTION\_TRACE field in the EXECUTION\_TRACES table. You might need a size qualifier if SQL raises exceptions with the <Lob-Value> reason code.

## Step 2: Deploying the Decision Validation Services archive for the first time

Deploy the Decision Validation Services archive for the first time.

### About this task

If you want to check the availability of Decision Validation Services, you must deploy the default SSP (Scenario Service Provider) EAR as described below.

**Note:** You must deploy the archive on the same server as the execution unit (XU).

### Procedure

Copy the JBoss file `<ODM_InstallDir>/executionserver/applicationservers/JBoss6EAP/jrules-ssp-JBOSS61EAP.ear` to `<JBoss_HOME>/server/standalone/deployments`.

You can use this procedure to deploy the default SSP archive packaged with the installer or any subsequent deployment of a repackaged archive.

If you have an XML XOM, you can also use the SSP EAR to test your rules.

**Note:** Because the ruleset archive inside a RuleApp includes an XML XOM, you do not have to repackage the SSP EAR to include the XOM.

## Step 3: Checking the availability of Decision Validation Services

A specific testing URL is designed for you to check the rule session type, the DAO factory class for trace persistence, the class used to cache the Decision Validation Services job, and the pool size for asynchronous execution.

To check the availability of Decision Validation Services:

1. Enter the URL `http://<host>:<port>/testing` in a web browser.
2. Log in to the (Scenario Service Provider) SSP application in one of the Rule Execution Server roles.

The application displays a home page, which contains information about the SSP server.

### Version

The version of Decision Server used.

### Patch level

The patch level of Decision Server used.

### License information

The type of license of this version.

### RuleSession

The rule session type (POJO or J2SE).

### DAO Factory Class

The DAO (Data Access Object) factory class that is used to persist the trace into the data warehouse.

### Job store class

The name of the class that is used to persist the Decision Validation Services job into a cache to free the memory during long computations.

### Job pool size

The size of the pool for asynchronous execution.

#### Started since

The time and date when the SSP started.

#### Jobs currently running

The About screen provides information about the jobs that are currently running after you run Decision Validation Services in Decision Center:

- A **Job ID** is listed in the table when a user clicks **Run** in Decision Center.
- The **Created** column records the date and time when each job is initialized.
- The **Status** column shows the number of scenarios that have already been tested, compared to the total number of scenarios.
- The **Start time** records the time when a resource is allocated for the job.
- The **Parts** column records the number of parts in the job:
  - A job that is not executed in parallel has one part.
  - A job that is executed in parallel has one or more parts.
- The **End time** records the time when the execution of the job is complete, that is, all of the scenarios in the job have been tested.

The report for the job is automatically downloaded by Decision Center at the end of the execution. If the scenario suite is run in the background, the user downloads the report by viewing the list of scenario suites, and then clicking the report link when it becomes available. After the report is viewed, the job is removed from the table. The job remains in the table until the report is downloaded.

## Step 4: Packaging Decision Validation Services by using Ant

You can configure Decision Validation Services archives by using an Ant task.

### About this task

This section is for users who install the product on Windows and other supported distributed platforms only.

You can configure Decision Validation Services archives by running the **ssp-setup** Ant task.

### Procedure

1. Define the Ant task in your build file by using the `<taskdef>` Ant element in one of the following ways:
  - Define the task at the top level, or within a specific target.

```
<taskdef resource="res-tasks.properties"
classpath="${<InstallDir>}/executionserver/lib/jrules-res-setup.jar"/>
```
  - If the JAR file is available in your system, you can write the following code:

```
<taskdef resource="res-tasks.properties"/>
```
2. Use the **ssp-setup** Ant task to update a Scenario Service Provider (SSP) artifact to your specific configuration and XOM.
3. Run the Ant task in one of the following ways:
  - From the command line: Execute Ant in the appropriate directory, followed by the name of the build file if necessary.
  - From Eclipse: Right-click the Ant file and click **Run**.

## **What to do next**

You can now deploy the testing archive.

---

## Chapter 2. Verifying your configuration of Decision Center

You can verify that you have correctly configured Decision Center by publishing some projects, opening the consoles, and running the diagnostics.

---

### Publishing a project to Decision Center

After completing the configuration, Decision Center is ready to be used but does not contain any rule projects. You publish rule projects from Rule Designer.

#### About this task

To publish a rule project to Decision Center, the project must be imported into your workspace.

The procedure uses the Decision Center tutorial projects as an example of how to import and publish a rule project. If you want to carry out the Decision Center tutorials, you have to publish the following projects:

- loanvalidation-rules (with loanvalidation-xom)
- loanvalidation-rules-dependent
- squery-loanvalidation-rules (with squery-loanvalidation-xom)

#### Procedure

1. To open Rule Designer, click **Start > All Programs > IBM > *package\_group* > Rule Designer**.
2. In Rule Designer click **File > Import > General > Existing Projects into Workspace**, and click **Next**.
3. Click **Select root directory**, browse to <InstallDir>/studio/tutorials/shared, and click **OK**.
4. Select the projects and click **Finish**.
5. Right-click the loanvalidation-rules rule project, and click **Decision Center > Connect**.
6. Complete the Decision Center Configuration dialog as follows. The warning message Connection not established displays until you establish the connection.

**User name**

rtsAdmin

**Password**

rtsAdmin

**URL** http://localhost:<port>/teamserver

**Data source**

Leave this field empty.

**Note:** If security is enabled, use https://localhost:<PORT\_NUMBER>/teamserver

7. Click **Connect**.

The connection is established when the warning message closes and the **Project configuration** area becomes active.

8. In the **Project configuration** area, check that **Create a new project on Decision Center** is selected, and then click **Finish**.
9. The Synchronize Complete - Decision Center Participant dialog opens when the publishing process is complete. Click **OK** to close this dialog.
10. A dialog opens asking you if you want to change to Team Synchronizing perspective. Click **Yes**.  
An empty Synchronize view opens, indicating that the projects in Rule Designer and Decision Center are the same. This means that your rules are now published to Decision Center.
11. Repeat for the other rule projects.

## What to do next

You can now open the Decision Center Enterprise console and perform diagnostics.

---

## Opening the Decision Center consoles

After you have deployed the Decision Center EAR or WAR to your application server, you can open the Decision Center consoles.

You can open the consoles by using the following URLs in a web browser:

- **Enterprise console:** `http://localhost:<PORT_NUMBER>/teamserver`
- **Business console:** `http://localhost:<PORT_NUMBER>/decisioncenter`

**Note:** If your browser is not running on the same host as the application server, replace `localhost` with the address of the machine. If your web application is mapped to a host with a port that is different from the default port, use the port number of the host.

By default, the data source is `jdbc/ilogDataSource`. If you want to specify a different data source, you have to pass it as a request parameter in the URL. For example:

```
http://localhost:7001/teamserver?datasource=jdbc/serverextendedbrm.
```

The locale of the sign-in page is English by default. You can specify a locale parameter in the URL that switches the sign-in page to the required locale. For example:

```
http://localhost:<port>/teamserver?locale=es (assuming that your message files are localized).
```

If you sign in with another locale in the URL and want to change the locale afterward, click **Options** in the top banner of the Enterprise console or **Profile** in the Business console. This saves the locale and restores it the next time you sign in.

If you open Decision Center but no database exists, you automatically access the Installation Settings wizard with only the **Install** tab available.

After completing the installation, Decision Center is ready to use but does not contain a rule project. You have to publish a rule project from Rule Designer.

A diagnostics tool, available in the Configure tab of the Enterprise Console, shows a report on the status of your Decision Center configuration.



To learn more about Decision Center, see [Decision Center](#).



---

## Notices

This information was developed for products and services offered in the U.S.A.

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

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

IBM United Kingdom Laboratories,  
Mail Point 151,  
Hursley Park,  
Winchester,  
Hampshire,  
England SO21 2JN

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

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

**The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:**

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

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

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those

websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

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

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

IBM United Kingdom Laboratories,  
Mail Point 151,  
Hursley Park,  
Winchester,  
Hampshire,  
England SO21 2JN

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

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

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

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

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

#### COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or

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

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

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. \_enter the year or years\_.

---

## Trademarks

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

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

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.

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



---

# Index

## A

- Ant tasks
  - configuring Decision Center 47, 74
  - creating the database schema 50, 77
  - defining and uploading message files 53, 80
  - execute-schema 51, 78
  - executing the database schema 51, 78
  - gen-create-schema 50, 77
  - gen-drop-schema 52, 79
  - print-config-param 54, 81
  - remove-config-param 54, 81
  - removing a database schema 52, 79
  - repackage-war 56, 83
  - repackaging the Decision Center archive 56, 83
  - set-config-param 54, 81
  - setting configuration parameters 54, 81
  - setting the persistence locale 54, 81
  - setting up the environment 47, 74
  - ssp-setup 62, 89
  - to repackage an EAR file 14, 31
  - upload-extensions 51, 78
  - upload-messages 53, 80
  - upload-roles 51, 78
  - uploading groups to the database 51, 78
  - uploading the database schema extension 51, 78
- asynchronous execution
  - pool size 61, 88

## C

- cells
  - Rule Execution Server configuration in a cluster environment 14
- clusters
  - JBoss 16, 34
- configuration parameters
  - adding or removing 54, 81
  - setting up 46, 73
- configuring Decision Center
  - declaring custom groups 40, 67
  - deploying Decision Validation Services 61, 88
- configuring Decision Validation Services 62, 89
  - checking availability of Decision Validation Services 61, 88
  - creating database resources 59, 86
- configuring Decision Validation Services on JBoss 58, 85
- configuring Rule Execution Server environments 14
- connection pool
  - creating 38, 65
- connection pools
  - creating 4

- custom groups, declaring
  - Decision Center 40, 67

## D

- data source
  - creating 38, 65
- data sources
  - creating 4, 22, 23
- database
  - establishing credentials 4, 20
- database resources
  - for Decision Validation Services 59, 86
- database schema
  - creating 50, 77
  - executing 51, 78
  - for Rule Execution Server
    - choosing, using Installation Settings wizard 10, 28
    - reviewing, using Installation Settings wizard 11, 28
  - removing 52, 79
  - uploading extension 51, 78
- databases
  - creating a schema from the console 8, 26
  - permissions
    - Decision Center 37, 64
  - populating with SQL scripts 12, 29
  - user permissions 3, 20
- Decision Center
  - adding groups 39, 66
  - adding users 39, 66
  - configuring for JBoss 35, 63
  - configuring security 39, 66
  - configuring the database 43, 70
  - configuring with Ant tasks 47, 74
  - deploying on JBoss 41, 68
  - Installation Settings wizard 42, 69
  - opening 92
  - repackaging the archive 56, 83
  - setting up configuration parameters 46, 73
  - setting up groups 45, 72
  - setting up message files 44, 71
  - setting up persistence locale 45, 72
  - verifying deployment 42, 69
- Decision Validation Services
  - checking availability 61, 88
  - configuring 58, 85
  - deploying 61, 88
  - packaging using Ant 62, 89
- Decision Warehouse
  - persistence setting for execution trace 2, 19
- declaring
  - custom groups in Decision Center 40, 67
- deploying
  - Decision Validation Services 61, 88

- deployment descriptor 4, 20
- diagnostics
  - running 13, 31
- dropping a database schema
  - See removing

## E

- EAR archives
  - repackaging using Ant 14, 31
- environment variables
  - for Ant 47, 74
- environments
  - Rule Execution Server configuration 14

## G

- groups
  - adding 39, 66
  - adding in JBoss 6, 21
  - setting up 45, 72
  - uploading to the database 51, 78

## H

- hosted transparent decision service
  - deploying on JBoss 13, 30

## I

- Installation Settings wizard 8, 26, 42, 69
  - choosing the database schema 10, 28
  - Configure Database 43, 70
  - report 11, 29
  - reviewing the database schema 11, 28
- Set Configuration Parameters 46, 73
- Set Persistence Locale 45, 72
- Set up Groups 45, 72
- Set up Message File 44, 71
- installing Decision Center
  - restricting user permissions 37, 64

## J

- JBoss application server
  - adding groups 6, 21
  - adding users 6, 21
  - clusters 16, 34
  - configuring for Decision Center 35, 63
  - configuring for Rule Execution Server 1, 18
  - configuring security 6, 21, 39, 66
  - creating a connection pool 38, 65
  - creating a data source 4, 22, 23, 38, 65
  - creating a database schema 8, 26

JBoss application server (*continued*)  
  deploying Rule Execution Server 8,  
  25  
  deploying the Decision Center  
  EAR 41, 68  
  deploying the XU RAR 7, 24  
  hosted transparent decision  
  service 13, 30  
  verifying deployment of Decision  
  Center 42, 69  
JDBC driver  
  creating 22

## M

message files  
  defining and uploading 53, 80  
  setting up 44, 71

## O

opening  
  Decision Center 92  
  Rule Execution Server console 9, 27

## P

packaging  
  Decision Validation Services 62, 89  
  Rule Execution Server archives using  
  Ant 14, 31  
permissions 39, 66  
  database access 3, 20  
  Decision Center installation 37, 64  
persistence  
  configuring in Rule Execution  
  Server 2, 19  
persistence locale  
  setting 54, 81  
  setting up 45, 72  
publishing rule projects  
  from Decision Center 91

## R

removing  
  database schema 52, 79  
Rule Execution Server  
  configuring  
  using Ant tasks 14, 31  
  configuring for JBoss 1, 18  
  configuring security 6, 21  
  creating a database schema 8, 26  
  deploying on cluster  
  environments 15, 33  
  deploying on JBoss 8, 25  
  deploying the hosted transparent  
  decision service 13, 30  
  deploying the XU RAR 7, 24  
  populating a database 12, 29  
  running diagnostics 13, 31  
  setting up the persistence mode 2, 19  
Rule Execution Server console  
  opening 9, 27

rule projects  
  publishing  
  from Decision Center 91

## S

Scenario Service Provider (SSP)  
  ssp-setup Ant task 62, 89  
security settings  
  configuring 6, 21  
SQL scripts  
  to remove a database schema 52, 79  
ssp-setup, Ant task 62, 89

## U

user permissions 3, 20  
users  
  adding 39, 66  
  adding in JBoss 6, 21

## W

WebLogic Server application server  
  verifying deployment of Decision  
  Center 42, 69

## X

XU RAR  
  deploying on JBoss 7, 24  
XUConnectionFactory  
  default JNDI name 14







Printed in USA