

IBM Operational Decision Manager
Version 8 Release 5

*Configuring Operational Decision
Manager on Tomcat*



Note

Before using this information and the product it supports, read the information in "Notices" on page 47.

This edition applies to version 8, release 5, modification 1 of Operational Decision Manager and to all subsequent releases and modifications until otherwise indicated in new editions.

© **Copyright IBM Corporation 2012, 2013.**

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 Tomcat 1

Configuring Rule Execution Server on Tomcat	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	3
Step 4: Deploying the management Rule Execution Server WAR	4
Step 5: Creating a schema for the Rule Execution Server database	6
Creating a database schema by using the Rule Execution Server console	6
Installation Settings wizard overview	6
Opening the Rule Execution Server console	7
Step 1: Welcome to the Installation Settings wizard	8
Step 2: Choose the database schema	8
Step 3: Review the database schema	9
Step 4: The Installation Settings wizard report	9
Creating a database schema by running SQL scripts	10
Step 6: Deploying the hosted transparent decision service	11
Step 7: Verifying the configuration	12
Optional configuration steps	13
Repackaging the Rule Execution Server archive using Ant	13
Configuring Rule Execution Server for different environments	13
Configuring the Decision Center consoles on Tomcat	14
Before you start	14
Step 1: Restricting database user permissions	15
Step 2: Creating a data source and connection pool	16
Step 3: Configuring security	17
Groups and permissions	17
Declaring custom groups	19
Step 4: Deploying the Decision Center WAR file	20
Step 5: Verifying the deployment of the Decision Center Enterprise console	20
Step 6: Completing Decision Center configuration	21

Completing the configuration from the Decision Center Enterprise console	21
Installation Settings wizard overview	21
Step 1: Configure the database	22
Step 2: Set up message files	22
Step 3: Set up groups	23
Step 4: Set the persistence locale	24
Step 5: Set configuration parameters	24
Completing the configuration by using Ant tasks	26
Setting up the Ant tasks environment	26
Creating the database schema	27
Defining and uploading message files	31
Setting the persistence locale	32
Adding or removing configuration parameters	32
Repackaging the Decision Center archive	34
Configuring the search function of the Decision Center Business console	35
Using a remote Solr search engine	35
Setting parameters for the Solr search engine	35
Additional steps to configure Decision Validation Services	36
Before you start	36
Step 1: Creating Decision Warehouse database resources	37
Step 2: Deploying the Decision Validation Services archive	39
Step 3: Checking the availability of Decision Validation Services	40
Step 4: Packaging Decision Validation Services by using Ant	41

Chapter 2. Verifying your configuration of Decision Center. 43

Publishing a project to Decision Center	43
Opening the Decision Center consoles	44

Notices	47
Trademarks	49

Index	51
------------------------	-----------

Chapter 1. Configuring Operational Decision Manager on Tomcat

Configuring Rule Execution Server on Tomcat

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

Operational Decision Manager 8.5.1 supports Tomcat 7

Before you start

To configure Rule Execution Server on Tomcat, 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.





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

Note: There are also some “Optional configuration steps” on page 13.

What steps to follow

The following table shows the steps that you must follow to configure Rule Execution Server on Tomcat 7.0, depending on the type of persistence 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 3	Not applicable		
“Step 4: Deploying the management Rule Execution Server WAR” on page 4			

Installation Steps		Persistence		
		File	Data source	JDBC
"Step 5: Creating a schema for the Rule Execution Server database" on page 6	"Creating a database schema by using the Rule Execution Server console" on page 6	Not applicable		
	"Creating a database schema by running SQL scripts" on page 10	Not applicable		
"Step 6: Deploying the hosted transparent decision service" on page 11		Optional	Optional	Optional
"Step 7: Verifying the configuration" on page 12		Recommended	Recommended	Recommended

Related concepts:

Troubleshooting Rule Execution Server on Tomcat

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/resetup.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: 5.1.6. Server SQL Modes and 5.1.3 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.

About this task

These credentials are required to establish the datasource, which you will do in the next step (“Step 4: Deploying the management Rule Execution Server WAR”).

If a database does not already exist, create one now by following the instructions for that database type. The procedure below shows how to create an empty Derby database. You can also create the database in the deployment step.

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

Procedure

1. Stop the application server.
2. Launch <DerbyInstallDir>/bin/ij.bat (or ij for Linux).
3. Create the database and connect to it.

For example, to create the new database c:/resdb as the user resdbUser and connect to it, use the command:

```
ij>connect 'jdbc:derby:c:/resdb;user=resdbUser;password=resdbUser;create=true';
```

4. Close the ij utility.

```
ij> quit;
```

What to do next

Now that you have created the database, you must create the database schema to contain its tables and views. You do this after deploying the WAR in the next step.

Note: On Tomcat, you use the execution unit (XU) as a Java SE component.

Step 4: Deploying the management Rule Execution Server WAR

Deploy the WAR file to use Rule Execution Server on Tomcat.

About this task

Also, 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

Use the following procedure to create groups and users, and deploy the Rule Execution Server console management WAR.

Note: For more information about how to set the JNDI name of the data source, see JNDI Datasource HOW-TO.

Procedure

1. Copy the file <InstallDir>/executionserver/applicationserver/tomcat7/res.war to {TOMCAT_HOME}/webapps.
2. Open {TOMCAT_HOME}/conf/tomcat-users.xml.
3. In the file, add the following code after the <tomcat-users>:

```
<role rolename="resAdministrators"/>
<role rolename="resDeployers"/>
<role rolename="resMonitors"/>
<user username="resAdmin" password="resAdmin" roles="resAdministrators,
resDeployers,resMonitors"/>
<user username="resDeployer" password="resDeployer"
roles="resDeployers,resMonitors"/>
<user username="resMonitor" password="resMonitor" roles="resMonitors"/>
```
4. Save the file.
5. Copy your database driver to the {TOMCAT_HOME}/lib directory. For example, the Derby embedded database driver is derby.jar, which is available in <InstallDir>/studio/samples/shared/lib.
6. Open {TOMCAT_HOME}/conf/server.xml.
7. Declare the database just before the tag closing the global resources definitions: </GlobalNamingResources>. For example:

```

<Resource name="jdbc/resdatasource" type="javax.sql.DataSource"
  auth="Container"
  username="resdbAdmin" password="resdbAdmin"
  driverClassName="org.apache.derby.jdbc.EmbeddedDriver"
  url="jdbc:derby:c:/resdb;create=true"/>

```

Note: Adjust the URL if you created the empty database in the previous step.

8. In the same file, add the link to your database for res.war just before the tag closing the localhost definition: </Host>:

```

<Context path="/res"
  docBase="res.war"
  debug="0"
  reloadable="true"
  crossContext="true">
<Logger className="org.apache.catalina.logger.FileLogger"
  prefix="localhost_res_log."
  suffix=".txt" timestamp="true"/>
<ResourceLink name="jdbc/resdatasource"
  global="jdbc/resdatasource"
  type="javax.sql.DataSource"/>
</Context>

```

9. Start Tomcat by launching startup.bat in {TOMCAT_HOME}/bin
10. Open the Rule Execution Server console in your browser by entering res with 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 with a different port number, change the port number from 8080 to the host port number.

The log for the Rule Execution Server console is written to the {TOMCAT_HOME}/logs/res-console.log file.

Step 5: Creating a schema for the Rule Execution Server database

After you have created an empty database, you can create the schema for the Rule Execution Server 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 run the database scripts from the Installation Settings wizard of the Rule Execution Server console if you work on Windows or other distributed platforms.

Installation Settings wizard overview:

On Windows and distributed platforms, 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 or jdbc
Java XOMs	file	No Installation Settings wizard	RuleApps persistence details only
	datasource or jdbc	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 your database if you are using data source persistence or you set persistence to JDBC.
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 with a different port number, change the port number from 8080 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 are using 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 an empty database schema, and datasource or jdbc as the persistence setting, 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 10.

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 “resdb”

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 name of the script that creates the database schema is `repository_<DatabaseName>.sql`. By default, the script is in `<InstallDir>/executionserver/databases`.

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 have set Java XOM persistence to a database, you must create these tables by running the `xomrepository_<DatabaseName>.sql` script.

Use any tool that can handle SQL to import and run the SQL scripts. The following table lists the tools that are provided for each database.

Table 1. Database scripting tools

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 secure the following credentials and privileges.

- The database user name and password
- Complete privileges on the tables, and a view of the schema (create, insert, delete)
- *Create index* privileges
- On Oracle, *create trigger* and *create sequence* privileges

Install a database client for the database that you use, and refer to the documentation of your database system for more information. The following example assumes that you have installed the Derby database.

Example: Running the Derby SQL scripts

This example assumes that you have created an empty Derby database in step 3 and deployed the WAR in step 4.

To create the database schema in Derby:

1. Stop Tomcat by clicking `shutdown.bat` in `{TOMCAT_HOME}/bin`
2. Launch **<Derby_InstallDir>/bin/ij.bat**.
3. Connect to your existing database. For example, to connect to the database `c:/resdb` as the user `resdbUser`, use the command:
`ij> connect 'jdbc:derby:c:/resdb;user=resdbUser;password=resdbUser';`
4. Run the script that creates the database schema:
`ij> run '<InstallDir>/executionserver/databases/repository_derby.sql'`;
If the script is being run for the first time, some errors related to the drop statements might occur.

Note:

If you have installed Decision Validation Services, you must also create the required database schema by running the script `trace_derby.sql`. See Additional steps to configure Decision Validation Services for more information.

5. Close the `ij` utility:
`ij> quit;`
6. Start Tomcat by clicking `startup.bat` in `{TOMCAT_HOME}/bin`

Step 6: Deploying the hosted transparent decision service

You can optionally deploy the WAR file for hosted transparent decision services.

About this task

On Tomcat, you use the execution unit (XU) as a Java SE component. Use the following procedure to deploy the hosted transparent decision service WAR on Tomcat 7.0:

Procedure

1. Stop Tomcat by clicking `shutdown.bat` in `{TOMCAT_HOME}/bin`
2. Copy `<InstallDir>/executionserver/applicationservers/tomcat7/DecisionService.war` to `{TOMCAT_HOME}/webapps`.
3. In `{TOMCAT_HOME}/conf/server.xml`, add the following code just before the `</Host>` tag closing the local host definition:

```
<Context path="/DecisionService" docBase="DecisionService.war" debug="0"
reloadable="true" crossContext="true">
  <Logger className="org.apache.catalina.logger.FileLogger"
prefix="localhost_htds_log." suffix=".txt" timestamp="true"/>
  <ResourceLink name="jdbc/resdatasource" global="jdbc/resdatasource"
type="javax.sql.DataSource"/>
</Context>
```
4. Save the file.
5. If you use virtual context roots, set the `ilog.rules.res.HTDS_CONTEXT` parameter to the actual context root that maps to the physical hosted transparent decision

service application. In the WEB-INF/web.xml of the Rule Execution Server console, change the `<param-value>/DecisionService</param-value>` in the following `<context-param>` section:

```
<context-param>
<description>Hosted Transparent Decision Services context-root.
  Can be modified.</description>
<param-name>ilog.rules.res.HTDS_CONTEXT</param-name>
<param-value>/DecisionService</param-value>
</context-param>
```

For example, you could change `<param-value>/DecisionService</param-value>` to `<param-value>/myservice</param-value>`.

- Restart Tomcat and the Rule Execution Server console for the hosted transparent decision service to function correctly.
- Set the `ruleset.xmlDocumentDriverPool.maxSize` ruleset property to the appropriate value.
See Setting the `ruleset.xmlDocumentDriverPool.maxSize` property.

What to do next

For information about checking whether you have deployed the hosted transparent decision service successfully, refer to the Rule Execution Server console online help.

Step 7: Verifying the configuration

You can run diagnostics from the console to verify that you have successfully configured Rule Execution Server.

Procedure

To run the Rule Execution Server diagnostics:


- Open the Rule Execution Server console by typing `res` with 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 with a different port number, change the port number from `8080` to the host port number.

- Sign in to the Rule Execution Server console.
- Click the **Diagnostics** tab.
- Click **Run Diagnostics**.
A report shows all the diagnostic tests that have been run.
- Click **Expand All** to show the details for all the tests.

Results

The tests show **XU Lookup** and **XU MBean** in a warning state (). This warning is normal because there is no JNDI name to use with the Java SE execution unit (XU). The diagnostics do not execute the steps **RuleSession Execution** and **RuleSession Second Execution** because they depend on **XU Lookup**.

Note:

The XU MBean becomes visible after the first execution. More precisely, when a connection is requested, it creates the XU connector and the XU MBean. The connection can be a management connection, a ruleset execution connection, or an untyped connection to initialize the pool to a minimal number of connections, or to test the availability of the resource adapter.

Optional configuration steps

When configuring Rule Execution Server for the Tomcat application server, you can also repackage the management archive file with a different persistence setting or configure Rule Execution Server for different environments.

Repackaging the Rule Execution Server archive using Ant

When you need to repackage a Rule Execution Server archive to configure Rule Execution Server, you can use an Ant task if 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. For more information, see [Setting up your environment to use Ant](#).

About this task

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

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

Procedure

Write the code that creates a management `.war` file that sets the new persistence mode:

```
ant
-Dconsole.war.in=<ODM_INST>/executionserver/applicationservers/tomcat7/res.war
-Dconsole.war.out=mymanagement.war
-Dpersistence.type=file -f ressetup.xml setup
```

Configuring Rule Execution Server for different environments

General configuration guidelines apply if you want to enable Rule Execution Server on different environments in a single cell.

Before you begin

These instructions assume that you do not configure multiple instances of Rule Execution Server in a single cell.

About this task

Most likely, the development of your business rule management system (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: it includes 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. When 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.

Consider the following guidelines to set up your different environments in a single cell.

Procedure

1. Set up different data sources.
2. Deploy and configure a XU for each environment.
3. Deploy the Rule Execution Server console for each environment.
4. To set up a data source for each environment, use unique JNDI names.
For example: jdbc/resdatasourceEnv1 and jdbc/resdatasourceEnv2
5. To deploy the Rule Execution Server console for each environment, proceed as follows:
 - a. Modify the deployment descriptor of the Rule Execution Server EAR or WAR management archive: in the web.xml file, uncomment the **JMX_XU_QUERY_PART** parameter and specify xuName=xuEnv1.
 - b. Deploy the Rule Execution Server EAR or WAR file to the server.
In the resource, reference settings in the application server:
 - 1) Set the JNDI for the data source as:jdbc/resdatasourceEnv1.
 - 2) Set the JNDI name for the XU a: eis/ConnectionFactoryEnv1.
 - c. Repeat a and b to deploy the Rule Execution Server console for the other environments.
6. Restart the node agents after you complete the configuration.
7. Call the XU instances to register the XU with the Rule Execution Server console.

Configuring the Decision Center consoles on Tomcat

To be able to use the Decision Center Enterprise console on a new instance of Tomcat, you must deploy the provided archives for this server, and perform a number of configuration tasks.

Operational Decision Manager 8.5.1 supports Tomcat 7

Before you start

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

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

Before you start, do the following:

1. Delete any previous teamserv installations in the /temp and /webapps directories under <TOMCAT_HOME>. This ensures that Tomcat recompiles the JavaServer Pages.
2. Set TOMCAT_HOME as an environment variable.
3. Set JRE_HOME or JAVA_HOME as an environment variable.

After you have prepared your environment, refer to “What steps to follow” for the steps that you must complete to configure the Decision Center console for your application server.






If you need to configure Decision Validation Services, see “Additional steps to configure Decision Validation Services” on page 36.

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

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 following table lists the steps for configuring Decision Center on Tomcat 7.0.

Step		Mandatory/optional
“Step 1: Restricting database user permissions”		
“Step 2: Creating a data source and connection pool” on page 16		
“Step 3: Configuring security” on page 17	“Groups and permissions” on page 17	
	“Declaring custom groups” on page 19	Optional
“Step 4: Deploying the Decision Center WAR file” on page 20		
“Step 5: Verifying the deployment of the Decision Center Enterprise console” on page 20		Recommended
“Step 6: Completing Decision Center configuration” on page 21	“Completing the configuration from the Decision Center Enterprise console” on page 21	 Use one of the two methods to complete the installation.
	“Completing the configuration by using Ant tasks” on page 26	

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 used by Decision Center.

Before you begin

You must have a database to create a data source.

Note: Beginners using the Derby embedded database can create the database as part of the procedure below.

For a list of supported databases, consult the platform support details available on the IBM support site.

About this task

To create a data source and a connection pool for Decision Center on Tomcat, you copy the driver for your database to the Tomcat directory, and then you declare the database and link it to the Decision Center archive.

Procedure

1. Copy your database driver JAR files to the <TOMCAT_HOME>/lib directory. For example, the Derby embedded database driver is derby.jar, which is available in <InstallDir>/studio/samples/shared/lib.
2. Open <TOMCAT_HOME>/conf/server.xml.
3. In the file, declare the database and link it to teamserver.war by adding the following code just before the </Host> tag that closes the localhost definition, and then adjusting the properties:

```
<Context docBase="teamserver.war" path="/teamserver" reloadable="true">
    <Resource name="jdbc/ilogDataSource" auth="Container"
        type="javax.sql.DataSource"
        username="dbUser" password="dbPassword"
        driverClassName="driverClassName"
        url="URL" />
</Context>
```

4. Complete the common properties for local transaction data sources:

username and password

The tags username="dbUser" and password="dbPassword" are used to access the database. This user name and password is only used to access the database.

driverClassName

The class that is used to handle the connections to the database. For example, for the Derby embedded database:

```
driverClassName="org.apache.derby.jdbc.EmbeddedDriver"
```

URL

This provides the JDBC connection URL. For example:

```
url="jdbc:derby:c:/rtsdb;create=true"
```

In this example, you create a database named rtsdb on the local C: drive the first time you open the Decision Center console.

Note: You may need to configure datasource isolation to provide better concurrency experience, since Decision Center may encounter various database dead-locks if the transaction isolation level is not set to 'READ-COMMITTED'.

5. Save the file.

Step 3: Configuring security

To configure security for Decision Center on the Tomcat application server, you create user groups and define permissions for them. You can also create custom groups, provided that you declare them in the list of Tomcat users and upload them to the database.

Groups and permissions

You define user names, passwords, and groups.

Every user of Decision Center must belong to at least one of the mandatory groups: `rtsAdministrator`, `rtsConfigManager`, `rtsInstaller`, and `rtsUser`. These groups determine which parts of Decision Center a user can access.

You must create these groups for Tomcat. For testing purposes, use the suggested configuration below to create a default user name and password for each group.

The following table shows the available groups and descriptions.

Table 2. Decision Center permission groups

Group	Use
<code>rtsAdministrator</code>	Mandatory, gives the user access as an administrator.
<code>rtsInstaller</code>	Mandatory, gives the user access to the Installation Settings wizard.
<code>rtsConfigManager</code>	Mandatory, gives the user configuration manager access.
<code>rtsUser</code>	Mandatory, gives the user standard access.
Eligibility	Optional custom group, used in the Decision Center permissions tutorial: Tutorial: Implementing permissions
Validator	Optional custom group, used in the Decision Center permissions tutorial: Tutorial: Implementing permissions

Note: See Groups for more in-depth information on the Decision Center groups.

Suggested configuration

The following table shows the permission groups, and gives an example of the default user name and password that you can assign to each group.

Table 3. An example of user name and password for each permission group

Group	Default user/password
<code>rtsAdministrator</code> , <code>rtsConfigManager</code> , <code>rtsUser</code>	<code>rtsAdmin/rtsAdmin</code> <code>rtsInstall/rtsInstall</code>
<code>rtsConfigManager</code> , <code>rtsUser</code>	<code>rtsConfig/rtsConfig</code>
<code>rtsUser</code>	<code>rtsUser1/rtsUser1</code>
Eligibility	Eli/Eli
Validator	Val/Val

Follow these steps to implement a suggested configuration with a default user for each of the basic groups.

1. Open the `<TOMCAT_HOME>/conf/tomcat-users.xml` file.
2. In the file, add the following code for roles and users to just before the closing tag `</tomcat-users>`:

```
<role rolename="rtsConfigManager"/>
<role rolename="rtsUser"/>
<role rolename="rtsAdministrator"/>
<role rolename="rtsInstaller"/>
<role rolename="Eligibility"/>
<role rolename="Validator"/>
<user username="rtsUser1" password="rtsUser1" roles="rtsUser"/>
<user username="rtsAdmin" password="rtsAdmin" roles="rtsUser,rtsInstaller,
```

```

rtsAdministrator"/>
<user username="rtsConfig" password="rtsConfig" roles="rtsUser,
rtsConfigManager"/>
<user username="rtsInstall" password="rtsInstall" roles="rtsUser,rtsInstaller"/>
<user username="Eli" password="Eli" roles="Eligibility,rtsUser"/>
<user username="Val" password="Val" roles="Validator,Eligibility,rtsUser"/>

```

3. Save the file.

The above example also includes custom groups Validator and Eligibility, and their users, required to follow the Decision Center Tutorial: Implementing permissions.

Note: When completing the configuration, you have to upload all groups except rtsAdministrator and rtsInstaller to the database. See Step 3: Set up groups.

Declaring custom groups

When you create custom groups, you must declare them by adding them to the file that lists Tomcat users.

About this task

The Decision Center WAR file references the basic groups rtsUser, rtsConfigManager, rtsAdministrator, and rtsInstaller.

To use custom groups with the Decision Center permissions mechanism, you must declare them as follows:

1. Add them to the <TOMCAT_HOME>/conf/tomcat-users.xml file.
2. Upload them to the database from the Installation Settings wizard.

Table 4. An example of custom permission groups, roles, and credentials

Group	Use	Default user/password	Roles
Validator	Optional custom group, used in the Decision Center permissions tutorial.	Val/Val	Validator, Eligibility, rtsUser
Eligibility	Optional custom group, used in the Decision Center permissions tutorial.	Eli/Eli	Eligibility, rtsUser

The following procedure shows how to create and declare the Validator and Eligibility custom groups for the Decision Center Tutorial: Implementing permissions.

Note: These two groups are already added if you followed the previous step.

Procedure

1. Open the <TOMCAT_HOME>/conf/tomcat-users.xml file.
2. In the file, add the custom groups just before the closing tag </tomcat-users>.

For example:

```

<role rolename="Eligibility"/>
<role rolename="Validator"/>
<user username="Eli" password="Eli" roles="Eligibility,rtsUser"/>
<user username="Val" password="Val" roles="Eligibility,Validator,rtsUser"/>

```

3. Save the file.
4. 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.

What to do next

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

Step 4: Deploying the Decision Center WAR file

You copy the Decision Center management archive to the web application deployment directory of Tomcat.

Procedure

Copy the file `<InstallDir>/teamserver/applicationservers/tomcat7/teamserver.war` to the `<TOMCAT_HOME>/webapps` directory.

Important: The process of deploying the Decision Center management WAR file sets the persistence locale. After you have saved a rule to the database, you **must not** change the persistence locale. If you want to install Decision Center in a language other than English, see Setting the persistence locale.

What to do next

You can now verify the deployment of the Decision Center console.

Step 5: Verifying the deployment of the Decision Center Enterprise console

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: `http://localhost:<PORT_NUMBER>/teamserver`
Set `<PORT_NUMBER>` to the port number of your web application. If your browser is not running on the same host as the application server, replace the `localhost` with the address of the host.
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 Decision Center configuration

After you have set database user permissions, created a data source, and deployed the Decision Center WAR file, you complete the configuration either from the Decision Center console or by running Ant tasks.

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 Enterprise console*.

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](#).

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.
<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

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."

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/defaulttextextensionmessages_<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
teamserver.<extractorValidator>.class	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.
teamserver.build.path	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"> 1. First, use this property with the name of the user who started the server as the root for the cache: <code><build.path>_<username></code>. 2. If the path 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>. 3. If the system property is not defined, use the server directory and add <code>rtscache</code>. For example, <code><server_dir>/rtscache_<username></code>.
teamserver.br1.verbalizers	Specify the list of locales for which a BAL verbalizer is defined.
teamserver.br1.verbalizer.<locale>	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 repackage 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. Uncompress 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 uncompressed 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."

You can set three parameters for the search engine:

Table 6. Search parameters

Parameter	Description
search.SearchProvider	This parameter takes one of the following values: <ul style="list-style-type: none"> • SolrEmbedded: Use this value to select the embedded Solr search engine. • SolrRemote: Use this value to select a remote instance of the Solr search engine.
search.SolrEmbeddedDataDir	Use this optional parameter to direct the index of the embedded version (SolrEmbedded) to a specific directory on the Decision Center.
search.SolrRemoteUrl	Use this parameter with SolrRemote 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
External server 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> .	Set the <code>preferences.properties</code> file as follows: <code>search.SearchProvider=SolrRemote</code> <code>search.SolrRemoteUrl=http://mysearchserver:8983/solr</code>
Embedded server In this example, you store the index in the <code>c:/temp/DC-SearchIndex</code> directory on your computer.	Set the <code>preferences.properties</code> file as follows: <code>search.SearchProvider=SolrEmbedded</code> <code>search.SolrEmbeddedDataDir=c:/temp/DC-SearchIndex</code>

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

To configure Decision Validation Services on Tomcat, you can create Decision Warehouse database resources, deploy the `testing.war` archive, and check the availability of the services. You can also run an Ant task to package Decision Validation Services.

Before you start

Before you deploy and configure Decision Validation Services, you must install the product modules and configure them for the application server.

Start by doing the following:

1. Install Rule Execution Server, Rule Designer, and Decision Center on the same computer. Decision Validation Services is added to these modules.
2. Download and run the Java EE add-on for your application server.
3. Follow the instructions to configure Rule Execution Server and Decision Center on the application server.


Refer to “What steps to follow” for the tasks you must complete.

Instead of completing the configuration steps listed in “What steps to follow,” you can use an Ant task to deploy and configure Decision Validation Services.

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

What steps to follow

The following table shows the configuration steps for Decision Validation Services:

Step	Required
<p>“Step 1: Creating Decision Warehouse database resources” Tip: If you used the Installation Manager to create database resources (see Populating a database by using the Rule Execution Server console), or you used the SQL scripts (see Populating a database by running SQL scripts), you do not have to perform this task.</p>	Optional
<p>“Step 2: Deploying the Decision Validation Services archive” on page 39 This archive is the default testing archive packaged with the Decision Validation Services installer.</p>	Optional. Do this step if you want to check the availability of the feature. You can also use the default archive if you have an XML XOM. In this case, you do not have to repackage the archive.
<p>“Step 3: Checking the availability of Decision Validation Services” on page 40</p>	
<p>“Step 4: Packaging Decision Validation Services by using Ant” on page 41</p>	Optional. You can use this task as an alternative method for configuring or modifying the Decision Validation Services installation. However, this step is mandatory when you want to run a test or simulation on a rule project that uses a Java XOM.

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

Database	Database tool
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

To deploy the Decision Validation Services archive, you must copy the archive file to the appropriate location, add the necessary link to your database, and then restart the application server.

About this task

To check that Decision Validation Services is available, you can deploy a specific archive, called *testing.war*, instead of the default archive. You must deploy it on the same server as the execution unit (XU). You can apply this procedure to deploy the default testing archive packaged with the installer or any subsequent deployment of a repackaged archive.

You can also deploy the default testing archive, and use it to test your rules if you have an XML XOM. The ruleset archive includes an XML XOM inside a RuleApp, so you do not have to repackage the archive to include the XOM.

Procedure

1. Copy the file `<InstallDir>/executionserver/applicationservers/tomcat7/testing.war` to `<TOMCAT_HOME>/webapps`.
2. Open the `<TOMCAT_HOME>/conf/server.xml` file.
3. In the file, add the following link to your database for *testing.war* just before the tag that closes the localhost definition `</Host>`:

```
<Context path="/testing"
  docBase="testing.war"
  debug="0"
  reloadable="true"
  crossContext="true">
<Logger className="org.apache.catalina.logger.FileLogger"
  prefix="localhost_res_log."
  suffix=".txt" timestamp="true"/>
```

```
<ResourceLink name="jdbc/resdatasource"
  global="jdbc/resdatasource"
  type="javax.sql.DataSource"/>
</Context>
```

4. Save the file.
5. If necessary, restart Tomcat and the Rule Execution Server console so that the Decision Validation Services function correctly.

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 (P0J0 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 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.

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 26
- creating the database schema 28
- defining and uploading message files 31
- execute-schema 29
- executing the database schema 29
- gen-create-schema 28
- gen-drop-schema 30
- print-config-param 33
- remove-config-param 33
- removing a database schema 30
- repackage-war 34
- repackaging the Decision Center archive 34
- set-config-param 33
- setting configuration parameters 33
- setting the persistence locale 32
- setting up the environment 26
- ssp-setup 41
- to repack an WAR file 13
- upload-extensions 29
- upload-messages 31
- upload-roles 30
- uploading groups to the database 30
- uploading the database schema extension 29

asynchronous execution

- pool size 40

C

cells 13

configuration parameters

- adding or removing 33
- setting up 24

configuring Decision Center

- configuring security 17
- creating a connection pool Tomcat 16
- creating a data source 16
- creating groups 18
- creating permissions 18
- custom groups 19
- deploying Decision Validation Services 39
- deploying the management WAR file 20

configuring Decision Validation Services 41

- checking availability of Decision Validation Services 40
- creating database resources 38
- on Tomcat 36

configuring Rule Execution Server

- creating a database schema
 - by running SQL scripts 10
- from the Installation Settings wizard 6

configuring Rule Execution Server

- (continued)
- creating a Derby database
 - on Tomcat 4
- diagnostics 12
- establishing database credentials 4
- on Tomcat
 - deploying the hosted transparent decision service 11
 - deploying the management stack 4
 - overview 1
 - various environments 13

connection pools

- in Decision Center
 - creating 16

custom groups

- Decision Center 19

D

data sources

- in Decision Center
 - creating 16

database credentials

- creating an empty database
 - on Tomcat 4

database resources

- for Decision Validation Services 38

database schema

- creating 28
- executing 29
- for Rule Execution Server
 - choosing, using Installation Settings wizard 8
 - reviewing, using Installation Settings wizard 9
- removing 30
- uploading extension 29

databases

- creating schema
 - by running SQL scripts 10
 - creating the schema
 - from the Installation Settings wizard 6
- permissions
 - Decision Center 15
 - user permissions 3

Decision Center

- configuring the database 22
- configuring with Ant tasks 26
- Installation Settings wizard 21
- opening 44
- repackaging the archive 34
- setting up configuration parameters 24
- setting up groups 23
- setting up message files 22
- setting up persistence locale 24
- verifying deployment 20

Decision Center console

- deploying 20

Decision Validation Services

- checking availability 40
- configuring
 - on Tomcat 36
- deploying 39
- packaging using Ant 41

Decision Warehouse

- persistence setting for execution trace 2

declaring

- custom groups in Decision Center 19

deploying

- Decision Center WAR file 20
- Decision Validation Services 39
- hosted transparent decision services
 - on Tomcat 11

Derby databases

- creating a database 4

diagnostics

- in the Rule Execution Server console
 - running 12

dropping a database schema

- See removing

E

environment variables

- for Ant 26
- to configure Rule Execution Server for Tomcat 1

environments

- Rule Execution Server
 - configuration 13

G

groups

- Decision Center 18, 19
- setting up 23
- uploading to the database 30

H

hosted transparent decision services

- deploying on Tomcat 11

I

Installation Settings wizard 1, 21

- choosing the database schema 8

Configure Database 22

report 10

- reviewing the database schema 9

Set Configuration Parameters 24

Set Persistence Locale 24

Set up Groups 23

Set up Message File 22

Installation Settings wizard (*continued*)
to create a Rule Execution Server
database schema 6
installing Decision Center
restricting user permissions 15

J

JBoss application server
verifying deployment of Decision
Center 20

M

management stack
deploying
on Tomcat 4
message files
defining and uploading 31
setting up 22

O

opening
Decision Center 44
Rule Execution Server console 7

P

packaging
Decision Validation Services 41
Rule Execution Server archives using
Ant 13
permissions
database access 3
Decision Center 18
Decision Center installation 15
persistence
configuring in Rule Execution
Server 2
persistence locale
setting 32
setting up 24
publishing rule projects
from Decision Center 43

R

removing
database schema 30
Rule Execution Server
configuring
using Ant tasks 13
diagnostics 12
setting up the persistence mode 2
Rule Execution Server console
deploying the management WAR on
Tomcat 4
opening 7
rule projects
publishing
from Decision Center 43

S

Scenario Service Provider (SSP)
ssp-setup Ant task 41
security
configuring
for Decision Center 17
SQL scripts
to create a database schema 10
to remove a database schema 30
ssp-setup, Ant task 41

T

Tomcat application server
database, empty 4

U

user permissions 3

W

WAR archives
repackaging using Ant 13
WAR files
deploying Decision Center 20
WebLogic Server application server
verifying deployment of Decision
Center 20

X

XUConnectionFactory
default JNDI name 13



Printed in USA